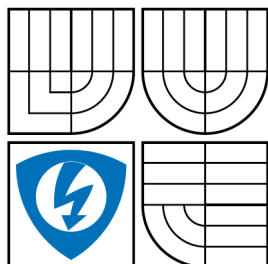


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# KOMUNIKAČNÍ TECHNOLOGI ZIGBEE V AUTOMATIZACI BUDOV

ZIGBEE IN BUILDING AUTOMATION

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. RADOVAN LIŠKA

VEDOUcí PRÁCE  
SUPERVISOR

Ing. PETR HONZÍK, Ph.D.

BRNO 2011



**VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky  
a komunikačních technologií**

**Ústav automatizace a měřicí techniky**

# Diplomová práce

magisterský navazující studijní obor  
**Kybernetika, automatizace a měření**

**Student:** Bc. Radovan Liška

**ID:** 73010

**Ročník:** 2

**Akademický rok:** 2010/2011

## NÁZEV TÉMATU:

**Komunikační technologi ZigBee v automatizaci budov**

## POKYNY PRO VYPRACOVÁNÍ:

Navrhněte infrastrukturu bezdrátové senzorové sítě pro sběr dat v budovách. V rámci návrhu vytvořte software pro funkci koordinátora bezdrátové sítě a pro směrovače bezdrátové sítě. Koordinátor by měl mít připojení na PC, s možností nastavení konfigurace a monitorování stavu sítě. Návrh demonstруйте na jednoduché síti, kde budou jednotlivé uzly předávat data dle konfigurace.

## DOPORUČENÁ LITERATURA:

- [1] CROSS, Pete. Zeroing on ZigBee. Circuit Cellar. [online]. 2005, iss. 175. Dostupné z <<http://www.circuitcellar.com/library/print/0205/Cross175/2.htm>>. ISSN 1528-0608.
- [2] STALLINGS William. Wireless Communications & Networks. 2.vyd. Upper Saddle River (NJ): Pearson Prentice Hall, 2005. ISBN 0-13-191835-4.
- [3] Atmel: AVR2050: BitCloud User Guide. [online]. 2009. User Guide. Dostupné z <[http://www.atmel.com/dyn/resources/prod\\_documents/doc8199.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8199.pdf)>

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 23.5.2011

**Vedoucí práce:** Ing. Petr Honzík, Ph.D.

**prof. Ing. Pavel Jura, CSc.**  
*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Dôsledkom napredovania vedy a techniky dochádza i k zlepšovaniu bezdrôtových technológií. Využitie bezdrôtových sietí založených na technológií ZigBee, ktoré sú už v dnešnej podobe schopné samostatného fungovania, poskytovania detailných informácií o fyzickom prostredí a riadenia procesov, prináša v sebe veľké množstvo výhod. Na druhej strane je v tejto novej technológii veľkou otázkou problém párovania zariadení. Práca sa zaoberá krátkym úvodom do sietí ZigBee, popisuje problematiku párovania Zariadení v ZigBee sieti a ich jednotlivé typy. Hlavnou časťou je vlastný návrh riešenia tohoto problému, navrhnutie algoritmu s využitím knižníc Bitcloudu pre Koordinátor, Koncové Zariadenie a Router a jeho demonštrovanie na vytvorenej aplikácii. Pri rozoberaní aplikácie sú popísané možné riešenia pre vytvorenie novej siete, pridanie nového uzlu do siete a tiež výmena zariadenia kus za kus. Výsledkom je grafická aplikácia a firmware pre jednotlivé zariadenia. Výsledok úspešného pripojenia sa zariadení do siete je podložený meraním.

## KĽÚČOVÉ SLOVÁ

bezdrôtové siete, ZigBee, párovanie zariadení, ZigBit, Adresovanie zariadení v sieti, Topológia siete, Bitcloud, IEEE 802.15.4, konfiguračný nástroj a aplikácia

## ABSTRACT

Improvement of wireless technologies is a natural consequence of the progress in the field of science and technology. Usage of wireless networks based on the ZigBee technology, which are capable of indepent operation in present form, providing detailed information about physical environment and process management, brings many advantages. On the other hand, there is a serious issue about commissioning. This Master's Thesis deals with introduction of ZigBee technology and its usage, describes issue about device commissioning and types of commissioning. The main part of the Thesis is my own proposal for solving this problem, proposing algorithm using the Bitcloud stack for Coordinator, End Device and Router and its demonstration at the application. Along with analysing the application there are described possible solutions for creating a new network, adding a new node into the existing network and changing a node. The result is a graphical application and firmware for each device. The result of succesfully associated devices in network is supported by the measurement.

## KEYWORDS

wireless networks, ZigBee, commissioning, ZigBit, Addressing, Network topology, Bitcloud, IEEE 802.15.4, commissioning tool

LIŠKA, Radovan *Komunikační technologi ZigBee v automatizaci budov*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2011. 63 s. Vedoucí práce byl Ing. Petr Honzík, Ph.D.

## PREHLÁSENIE

„Prehlasujem, že moju diplomovú prácu na tému „Komunikační technologi ZigBee v automatizaci budov“ som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce. Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákoníku č. 40/2009 Sb.“

Brno .....

.....

(podpis autora)

Ďakujem týmto vedúcemu diplomovej práce Ing. Petrovi Honzíkovi, Ph.D. a hlavne svojmu konzultantovi diplomovej práce Ing. Ondřeji Hynčicovi za cenné pripomienky a odborné rady, ktoré mi poskytli pri vypracovávaní mojej diplomovej práce.

# OBSAH

<b>Úvod</b>	<b>10</b>
<b>1 Popis Štandardu ZigBee z Aplikačného hľadiska</b>	<b>12</b>
1.1 Výmena dát . . . . .	12
1.2 Vlastné siete PAN . . . . .	13
1.2.1 Vysielacie Kanále . . . . .	13
1.2.2 Identifikátory siete . . . . .	13
1.2.3 Rozšírené Identifikátory siete . . . . .	14
1.3 Typy Zariadení v sieti . . . . .	14
1.4 Adresovanie v Sieti . . . . .	15
1.4.1 Sieťová Adresa Uzlu . . . . .	15
1.4.2 MAC Adresa Uzlu . . . . .	16
1.4.3 Adresovanie v Uzle . . . . .	16
1.4.4 Objekty Aplikačnej vrstvy . . . . .	18
<b>2 Spôsoby Párovania zariadení v ZigBee sieťach(Commissioning)</b>	<b>20</b>
2.1 Jednoduché Párovanie . . . . .	21
2.2 Komerčné párovanie . . . . .	23
2.3 Užívateľské párovanie . . . . .	24
2.4 Vyhodnotenie spôsobov párovania a výber typu pre aplikáciu . . . . .	25
2.5 Vlastné Riešenie Aplikácie . . . . .	26
<b>3 Vlastnosti Testovanej Siete</b>	<b>28</b>
3.1 Topológia Siete . . . . .	28
3.2 Formovanie Siete Star(Hviezda) . . . . .	29
<b>4 Návrh Aplikácie</b>	<b>31</b>
4.1 Firmware pre Jednotlivé Uzle Siete . . . . .	31
4.1.1 Návrh Algoritmu . . . . .	35
4.1.2 Implementácia Algoritmu pre Koordinátor . . . . .	36
4.1.3 Implementácia Algoritmu pre Koncové Zariadenie a Smerovač . . . . .	40
4.2 Návrh Softvéru . . . . .	41
4.2.1 Nízkoúrovňová Komunikácia a Protokol . . . . .	42
4.2.2 Popis Aplikačného Prostredia (GUI) . . . . .	45
4.3 Správa Siete z Pohľadu Užívateľa . . . . .	47
4.4 Výsledky časov pripojenia sa zariadení do siete pri výpadku . . . . .	48
<b>5 Záver</b>	<b>50</b>

<b>Literatura</b>	<b>52</b>
<b>Zoznam symbolov, veličín a skratiek</b>	<b>54</b>
<b>Zoznam príloh</b>	<b>57</b>
<b>A Stavový diagram ZC</b>	<b>58</b>
<b>B Stavový diagram ZED, alebo ZR</b>	<b>59</b>
<b>C Stavový diagram USART</b>	<b>60</b>
<b>D Vývojový diagram obsluhy USART</b>	<b>61</b>
<b>E Sekvenčný diagram pre zostavenie siete</b>	<b>62</b>
<b>F Obsah Priloženého CD</b>	<b>63</b>
F.0.1 adresár Firmware-C . . . . .	63
F.0.2 adresár Grafická Aplikácia-java . . . . .	63
F.0.3 zip súbor: BitCloud_ZIGBIT_1_10_0.zip . . . . .	63
F.0.4 pdf súbor: Diplomova_praca.pdf . . . . .	63

# ZOZNAM OBRÁZKOV

1.1	Mesh sieť - príklad využitia redundantných ciest pri prenose dát z A do B (selfhealing)[21] . . . . .	13
1.2	Rozdiel medzi štandardmi pri definovaní funkcií zariadení v sieti [8] . .	15
1.3	Adresácia vo vnútri zariadenia [20] . . . . .	17
2.1	Architektúra ZigBee/IEEE 802.15.4 vrstiev a objektov . . . . .	21
2.2	ZigBee jednoduché párovanie . . . . .	22
2.3	Komerčné párovanie s dvoma krokmi . . . . .	24
2.4	princíp procesu pripojenia sa do siete . . . . .	26
3.1	Príklad topológie typu hviezda a P2P . . . . .	28
3.2	Dátový tok v sieti . . . . .	30
4.1	Vývoj firmware do uzlov . . . . .	32
4.2	Vývojový set ZigBit 900[15] . . . . .	33
4.3	Architektúra BitCloud Software stacku . . . . .	34
4.4	Deployment diagram algoritmov bežiacich na rôznych zariadeniach . .	36
4.5	Hlavné súbory potrebné pre preklad projektu a výsledok prekladu . .	37
4.6	proces konfigurácie siete . . . . .	39
4.7	USART výmena dát v callback móde . . . . .	42
4.8	Class diagram Aplikačnej úlohy . . . . .	44
4.9	Grafické rozhranie aplikácie . . . . .	45
A.1	Stavový diagram pre Koordinátor . . . . .	58
B.1	Stavový diagram pre Koncové zariadenie a Smerovač . . . . .	59
C.1	Stavový diagram pre Komunikáciu zariadení s externým zariadením určeným na konfiguráciu . . . . .	60
D.1	Vývojový diagram obsluhy USART . . . . .	61
E.1	Sekvenčný diagram pre zostavenie siete . . . . .	62



# ZOZNAM TABULIEK

1.1	Adresovacie rozsahy v ZigBee . . . . .	16
2.1	Vyhodnotenie spôsobov párovania . . . . .	27
4.1	Tabuľka meraní doby pripojenia sa zariadení pri priamej viditeľnosti, vzdialenosť 20cm . . . . .	48
4.2	Tabuľka meraní doby pripojenia sa zariadení pri prekážke(hrúbka steny 15cm, vzdialenosť 5m) . . . . .	48
4.3	Tabuľka meraní doby pripojenia sa zariadení pri prekážke(hrúbka steny 2x15cm, vzdialenosť 15m) . . . . .	49
4.4	Tabuľka meraní doby pripojenia sa zariadení pri prekážke(hrúbka steny 15cm, vzdialenosť 25m) . . . . .	49

# ÚVOD

Technológia ZigBee sa začína implementovať do širokého spektra produktov, ktoré sú určené nielen pre domáce použitie, ale aj pre priemysel. Spomeňme aspoň zopár príkladov použitia : Osvetlenie priestorov, vyhrievanie, klimatizácie a vetranie, odber spotreby vody, plynu, závlahové systémy, monitorovanie zariadení a iné. Úlohou je zber nameraných dát zo snímačov, ktoré sú umiestnené v rozsiahlych priestoroch budov, či plôch, ich prenos a spracovanie s požadovaným výsledkom pre konkrétny typ aplikácie. V zdravotníckych strediskách môžeme pomocou ZigBee monitorovať zdravotný stav pacientov vrátane merania krvného tlaku či pohybu pacientov. Lekári alebo zdravotné sestry majú tak prehľad o svojich pacientoch bez ich zbytočného kontrolovania. I pomocou tejto možnosti sme schopný zabezpečovať väčší komfort pre pacientov. V domácej automatizácii je ZigBee prínosom pre väčší komfort a nižšie náklady pri prevádzkovaní zariadení. Vďaka bezdrôtovej komunikácii môžeme ľubovoľne meniť logické spojenie zariadení bez zásahu do hardware. Pomocou jedného tlačidla na diaľkovom ovládaní môžeme prepnúť požadované spotrebiče do režimu, ktorý máme nastavený. Ako príklad poslúži zatváranie okien, kedy nemusíme ručne zatvárať okno po okne, ale cez diaľkové ovládanie ich zavrieme naraz všetky.

Spomínané možnosti využitia sú obrovské. Avšak jednou z otvorených tém tejto technológie v porovnaní napríklad s Wi-Fi <sup>1</sup>, či Bluetooth [4], je otázka párovania (Commissioning), teda logického spojenia jednotlivých zariadení v sieti, hlavne z hľadiska užívateľskej náročnosti, teda ako obtiažné bude zistiť výpadok niektorého zariadenia a jeho opätovného uvedenia do prevádzky. Hlavným cieľom práce je poskytnutie možného riešenia tohto problému, konkrétne sa zameriavam na automatickú asociáciu zariadenia do siete v prípade, že už v sieti komunikoval, na konfiguráciu siete s použitím externého zariadenia - PC a na vytvorenie grafickej aplikácie do tohto zariadenia, pomocou ktorej nastavím parametre do jednotlivých zariadení.

Obsah dokumentu je členený do piatich kapitol. V prvej podávam obraz o použití štandardu ZigBee a jeho hlavné výhody. V druhej kapitole sú popísané možné spôsoby párovania, ich typy, zrovnanie a nakoniec i výber konkrétneho typu pre moju prácu. V tretej kapitole podrobne rozoberiem :

- navrhnutý algoritmus (firmware) pre tri hlavné typy zariadení - ZC (ZigBee koordinátor – ZigBee Coordinator), ZR (ZigBee smerovač – Router) a ZED (ZigBee koncové zariadenie – ZigBee End Device).

---

<sup>1</sup>Wi-Fi Aliancia založená v roku 1999 s cieľom bezproblémového prepojenia zariadení, obchodných segmentov a rozsiahlych oblastí[6]

- spôsob komunikácie medzi koordinátorom a pripojeným vzdialeným zariadením PC (počítač – Personal Computer), PDA (mobilné zariadenie – Personal Digital Assistant) ...
- Návrh aplikácie pre konfiguráciu a správu siete.
- úlohu užívateľa pri správe siete.

Súčasťou práce je tiež vyhodnotenie merania rýchlosti pripojenia uzlov ku koordinátorovi pri ich uvedení do siete - kapitola 4. V závere popisujem celkové zhodnotenie práce, možné vylepšenia a návrhy na pokračovanie v práci.

# 1 POPIS ŠTANDARDU ZIGBEE Z APLIKAČNÉHO HĽADISKA

## 1.1 Výmena dát

Hlavným poslaním bezdrôtových sietí je spoľahlivá výmena dát medzi uzlami v sieti. ZigBee sieť automaticky rozhoduje o výbere trasy pri výmene dát z jedného uzlu do druhého. ZigBee využíva mechanizmus pre prenos dát definovaný v IEEE. Mechanizmus zahŕňa:[10]

- Požiadavok na dáta - Data Request: ide o požiadavok na poslanie dát
- Potvrdenie prijatia dát - Data Confirm: znamená požiadavok na potvrdenie Požiadavku dát
- Indikácia dát - Data Indication: ide o indikáciu prijatých dát

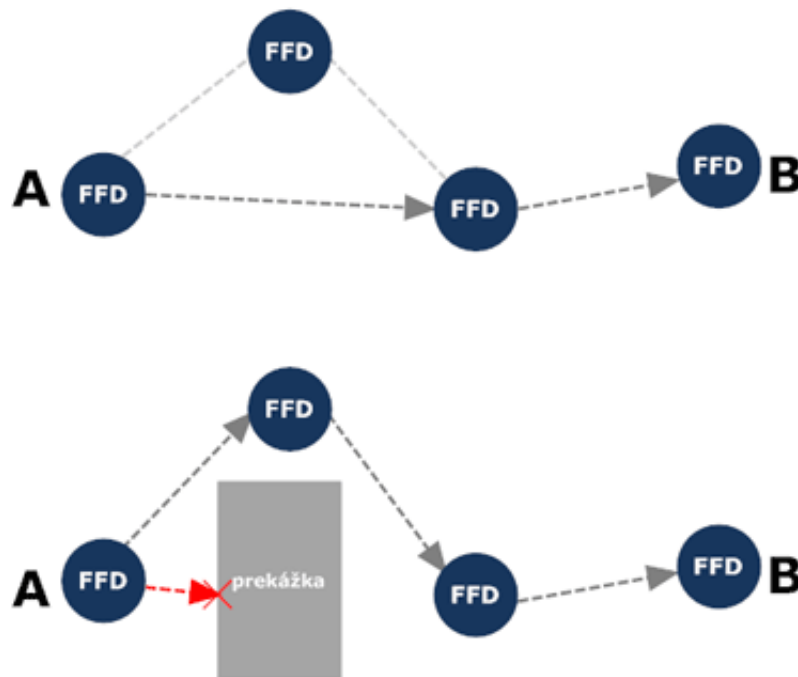
Požiadavka na dáta je generovaný aplikáciou. Pokaždé, keď aplikácia generuje Požiadavok na dáta, automaticky obdrží potvrdenie, ktoré indikuje doručené, alebo nedoručené dáta [13]. Dáta môžu byť doručené v ľubovoľnom čase. Požiadavok na dáta môže byť poslaný rôznymi možnosťami:

- Unicast s koncovým potvrdzovaním (end-to-end acknowledgement)
- Unicast bez koncového potvrdzovania
- Broadcast
- Groupcast/Multicast

Unicast znamená posielanie dát konkrétnemu uzlu. Broadcast je posielanie dát všetkým uzlom v sieti a Groupcast/Multicast je posielanie dát určitým uzlom v sieti, teda skupine [19]. podľa typu cez akú adresu posielame broadcast rozlišujeme:

- 0xffff : Broadcast všetkým zariadeniam v sieti
- 0xfffd : Broadcast všetkým zariadeniam, ktoré nie su v stave sleep
- 0xfffc : Broadcast všetkým smerovačom a koordinátorovi.

Pri ZigBee bezdrôtových sieťach sa využíva výhod *Mesh* topológie [17]. Základom je peer-to-peer model IEEE 802.15.4. O Mesh sa stará sieťová vrstva, ktorá je plne v súlade so ZigBee štandardom. Sieť je tvorená redundantným množstvom uzlov [23]. V prípade zlyhania niektorého z uzlov, prípadne pri vzniku prekážky sa hľadá automaticky nová trasa pre posielanie dát [14]. Príklad môžeme vidieť na obrázku č.1.1.



Obr. 1.1: Mesh sieť - príklad využitia redundantných ciest pri prenose dát z A do B (selfhealing)[21]

## 1.2 Vlastné siete PAN

ZigBee zariadenia tvoria sieť, ktorú nazývame PAN (Personal Area Network - Osobné siete). Výmena dát medzi uzlami je možná len keď sa nachádzajú v rovnakej sieti. PAN formuje len ZC. Ostatné zariadenia ako ZED a ZR sa do sformovanej siete len pripájajú. ZigBee sieť je charakteristická svojím PAN ID (identifikátorom siete), rozšíreným identifikátorom - extended PAN ID a tiež kanálom na ktorom vysiela.

### 1.2.1 Vysielacie Kanále

ZigBee využíva vysielacie kanálov tak ako ich definuje 802.15.4. V 2.4 GHz pásme sú tieto kanále číslované od 11 - 26. Kanále 0 - 10 tvoria subgigahertzové pásmo. V Európe je na frekvencii 868 MHZ. Nevýhoda tohto pásma je, že je veľmi úzke, a preto môžeme využívať len jedného kanálu. Výhoda je väčší dosah. V testovanej sieti používam frekvenčné pásmo 915 MHZ na kanáli 10.

### 1.2.2 Identifikátory siete

PAN ID sa používajú ako názov hovorí na rozlíšenie jednotlivých sietí. Toto umožňuje viacerým sieťam nachádzajúcich sa v tesnej blízkosti im vlastnej prevádzky bez

vzájomného zásahu jednej siete do druhej.

ZigBee PAN ID je 16-bitové číslo v rozsahu od 0x0000 do 0x3fff. Prípád kolízie z hľadiska výberu PAN ID sa preto v ZigBee sieti nepredpokladá. Možnosť výskytu dvoch rovnakých identifikátorov je možná, ale len v prípade inej fyzickej lokality a iného vysielacieho kanála.

### 1.2.3 Rozšírené Identifikátory siete

Tvoria ich 64-bitové unikátne čísla, ktoré identifikujú PAN. Vo všetkých prípadoch, okrem jedného v sieti prebieha komunikácia používaním 16-bitových PAN ID. Ide o tzv. *Beacon Response*, ktorý je vyvolaný na základe *Beacon Request*, ktorý obsahuje práve tento rozšírený identifikátor siete, kvôli pripojeniu zariadenia do siete. Pokaždé keď sa chce ZigBee uzol pripojiť do siete, vysiela beacon request. V zápätí na to očakáva odpoveď a vyberá na základe vnútorného algoritmu ten správny a pripája sa do siete.

## 1.3 Typy Zariadení v sieti

Štandard IEEE 802.15.4 pre senzorické siete definuje 2 základné typy zariadení v sieti. Jedná sa o FFD (full-function device) a RFD (reduced-function device). FFD zariadenie je schopné vykonávať všetky náležité funkcie, ktoré sú popísané v IEEE 802.15.4 štandarde. Môže tiež zaujať ľubovoľnú funkciu zariadenia v sieti. RFD má na druhej strane obmedzené možnosti, ako napríklad: Zariadenie FFD môže komunikovať s akýmkoľvek zariadením v sieti, pričom zariadenie RFD môže komunikovať len s FFD zariadením. RFD sa využívajú len pre jednoduché typy úloh, ako napríklad zapnúť/vypnúť vypínač. Pamäť RFD zariadení je obyčajne menšia ako FFD zariadení. Funkcia zariadení v sieti podľa štandardu:

- **IEEE 802.15.4:** FFD zariadenie môže vykonávať tri následovné úlohy: koordinátor, PAN koordinátor a obyčajné zariadenie. *koordinátor* je FFD zariadenie schopné prenosu správ. Pokiaľ je zariadenie hlavným správcom PAN, tak sa nazýva PAN ZC.
- **ZigBee:** V tomto štandarde je terminológia trochu iná, avšak podobná. ZC je u IEEE 802.15.4 PAN ZC. Smerovač(Router) je zariadenie na úrovni ZC a ZED je *koncové zariadenie*. ZED sú obyčajne najlacnejšie prvky v sieti.

Rozdiel v rozdelení týchto dvoch štandardov je znázornený na obrázku č. 1.2.

V ZigBee sieťach je hlavnou úlohou ZC založiť(sformovať) sieť. ZC je jediné zariadenie, ktoré môže sieť sformovať. Pokiaľ ide o zabezpečenie siete má ako jediný



Obr. 1.2: Rozdiel medzi štandardmi pri definovaní funkcií zariadení v sieti [8]

právo pridávať do siete ostatné zariadenia. Akonáhle je sieť vytvorená, tak ZC funguje už len ako smerovač. ZC nie je potrebný pre normálnu prevádzku siete. Je však nenahraditeľný pri pripájaní zariadenia do siete a tiež jeho opustenia siete. Na základe *Trust Center* určuje, či danému uzlu prideliť prístup do siete, alebo nie.

ZR rozširuje mesh sieť. Zväčšuje jej rozsah a zvyšuje spoľahlivosť siete. ZR tak ako ZC preposiela pakety a dovoľuje zariadeniam sa pripojiť do siete.

ZED má veľké využitie pri prevádzke, kde nie je vhodný zdroj napájania sieťový adaptér, ale skôr batérie. Veľmi často sa využíva vlastnosť uspania zariadenia na určitý čas a následne jeho prebudenia. ZED môže byť v stave *RxOnIdle* true, alebo false. Pri *RxOnIdle* = false ZED môže spať rôzne dlhú dobu. V ZigBee je tento časový limit však definovaný niektorými aplikačnými profilmi. Po zobudení sa pýta ZC či má pre neho nejakú správu a následne sa zas uspáva. Pr opačnej možnosti *RxOnIdle* = true, ZED prijíma správy okamžite a okamžite na ne reaguje [8].

## 1.4 Adresovanie v Sieti

Rozsahy adries pre jednotlivé položky sú znázornené v tabuľke č.1.1.

### 1.4.1 Sieťová Adresa Uzlu

Sieťová adresa sa tiež nazýva *NwkAddr*, alebo skrátená adresa (short address). Pozostáva zo 16-bitového unikátneho čísla. V ZigBee sieti má ZC vždy adresu 0x0000. Sieťovú adresu získavajú uzly z daného rozsahu, uvedeného v tabuľke č.1.1 náhodne.

Názov	Rozsah	Popis
<i>Kanál</i>	11 - 26	Fyzická časť RF spektra
<i>PAN ID</i>	0x0000 - 0x3fff	Adresa siete v komunikačnom kanále
<i>NwkAddr</i>	0x0000 - 0xffff7	Adresa uzlu v sieti
<i>Endpoint</i>	1 - 240	Adresa aplikácie v uzle
<i>Cluster</i>	0x0000 - 0xffff	Objekt v aplikácii
<i>Command</i>	0x00 - 0xff	príkaz na vykonie v úlohe
<i>Attribute</i>	0x0000 - 0xffff	poradie dát v Clusteri

Tab. 1.1: Adresovacie rozsahy v ZigBee

### 1.4.2 MAC Adresa Uzlu

Akronym pre túto adresu je tiež IEEE adresa, alebo dlhá adresa (long address), prípadne rozšírená adresa (extended address). Je tvorená 64-bitovým číslom, ktoré jednoznačne identifikuje daný uzol - zariadenie na celom svete. Prvých 24 bitov zhora tvorí identifikátor organizácie - OUI [12], spodných 40-bitov je rezervovaných výrobcami dosiek - OEM.

Sieťová adresa a a MAC adresa nemajú žiadnu väzbu so sebou. Po odpojení uzlu zo siete a po jeho opätovnom prihlásení sa do siete, MAC adresa zostane tá istá, ale sieťová adresa sa obvyčajne zmení.

### 1.4.3 Adresovanie v Uzle

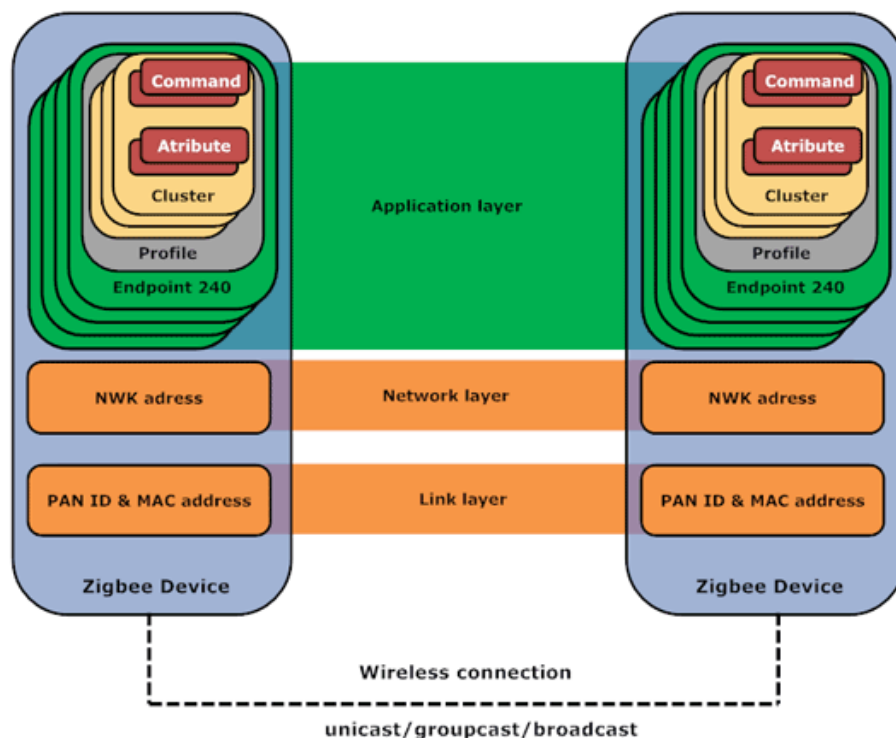
Ak poznáme sieťovú adresu zariadenia, sme schopný pomocou adresovacích metód typu broadcast, unicast a iné, doručiť dáta do správneho zariadenia. V prípade, že máme komplexnú sieť tvorenú uzlami, ktoré vykonávajú úplné odlišné funkcie, je treba rozlíšiť, ktorú z nich treba vybrať a vykonať. Na to nám slúži adresácia vo vnútri zariadenia. Zahŕňa nasledovné komponenty:

- PAN ID (MAC)
- NwkAddr (NWK)
- EndPoint (APS)
- Profil ID (APS)
- Cluster (APS)
- príkazy a atribúty (command / attribute) (ZCL)

Hierarchické rozloženie jednotlivých komponent v modeli je znázornené na obrázku č.1.3.

PAN ID môžeme vidieť na linkovej vrstve, sieťovú adresu na sieťovej vrstve a cieľový endpoint, profil ID, cluster ID a príkazy sú zahrnuté do aplikačnej vrstvy.





Obr. 1.3: Adresácia vo vnútri zariadenia [20]

**Endpoint:** je identifikátor pre aplikáciu vo vnútri uzlu. V jednom uzle môže bežať viacero aplikácií. Endpointy si môžeme predstaviť ako virtuálne spojenia aplikácií. Umožňuje rozličným zariadeniam a profilom v uzle navzájom fungovať.

**Cluster:** je aplikačný objekt, ktorý je definovaný 16-bitovým identifikátorom a obsahuje množinu atribútov a príkazov. Cluster definuje čo má daná aplikácia vykonať. Príklad môže byť OnOffCluster (ID 0x0006). Úlohou tohto clustra je len niečo zapínať, alebo vypínať. Nezáleží na aplikácii, ktorá to vyžaduje. Cluster zahŕňa kód - príkazy a dáta - atribúty.

#### Profily:

Požiadavky v ZigBee sú posielané a prijímané prostredníctvom aplikačných profilov. ID aplikačného profilu je 16-bitové číslo z rozsahu 0x0000 - 0x7fff pre verejné profily a 0xbfff - 0xffff pre špecifické profily výrobcov. Inými slovami povedané aplikačné profily slúžia na zjednotenie aplikácií od rôznych výrobcov. Napríklad termostat od firmy Honeywell môže spolupracovať s ovládacím ventilom od firmy Trane.

- *Verejné profily* :sú definované ZigBee Alianciou. Medzi verejné profily patrí napríklad profil HA (Home Automation), ktorý definuje široké spektrum zariadení určených pre domáce využitie ako napríklad: ovládače, svetlá a vypínače,

termostaty a iné. V jednej aplikácii môže zaraz existovať niekoľko aplikačných profilov.

- *Privátne profily* :sú definované výrobcami OEM.

Kompletný popis adresovania sa v aplikácii nastavuje pomocou *simple descriptor*. Prehľad jednotlivých parametrov a štruktúru simple Descriptor môžeme vidieť v časti kódu č.1.1.

Alg. 1.1: simpleDescriptor

```
Endpoint_t  endpoint
ProfileId_t  AppProfileId
uint16_t    AppDeviceId
uint8_t     AppDeviceVersion: 4
uint8_t     Reserved: 4
uint8_t     AppInClustersCount
ClusterId_t * AppInClustersList
uint8_t     AppOutClustersCount
ClusterId_t * AppOutClustersList
```

**APS:** Vrstva APS prekladá požiadavky aplikačnej vrstvy a poskytuje podklady spodných vrstiev pre aplikáciu. APS rámec obsahuje endpointy, clustre, profily ID a tiež skupiny. APS je zodpovedná za vykonávanie týchto činností: strana 193 v[10].

- filtrovanie paketov, pokiaľ endpoint nie je registrovaný, prípadne pokiaľ sa nezhodujú aplikačné profily.
- generovanie koncových potvrdení s opakovaniami.
- udržiavanie smerovacej tabuľky (binding table)
- udžiavanie lokálnych skupín v tabuľke
- udžiavanie lokálnych adresových máp

#### 1.4.4 Objekty Aplikačnej vrstvy

ZigBee obsahuje 2 sady služieb pre párovanie zariadení (commissioning) a udržiavanie siete v prevádzke. Sú to

- ZDO spolu s ZDP
- ZCL

ZDO je aplikácia bežiaca na enpoint 0. Poskytuje interfejs do ZDP - špeciálny aplikačný profil s ID 0x0000, ktorý slúži na objavovanie, konfigurovanie a udržovanie sieťovej prevádzky. Taktiež zasahuje do sieťovej vrstvy a rozhoduje, kedy sformovať sieť, alebo sa do siete pripojiť. V aplikácii tento objekt využívam na sformovanie siete pomocou funkcie *ZDO\_StartNetworkReq*. V iných aplikáciach je možnosť využitia funkcie pre tzv *sleep mode*, kedy zariadenia pracujú len v určitých časových intervaloch.

ZDP je ďalší objekt aplikačnej vrstvy, ktorého služby sú rozdelené do typu klient/server. Na strane klienta sú požiadavky(requests) a na strane servera odpovede na konkrétne požiadavky (responses). ZDP obsahuje nasledovné služby :

- Device Discovery - objavovanie nových zariadení
- Service Discovery - objavovanie aplikačných služieb bežiacich na vzdialenom zariadení
- Binding - párovanie : umožňuje tzv.nepriame adresovanie. Tento typ párovania poskytuje mechanizmus na prepojenie enpointov medzi sebou. Výmena dát je založená na existencii tabuľky, ktorá obsahuje informácie o : zdrojovom a cieľovom endpointe, MAC adresy, NWK adresy , Cluster ID a prípadne aj informáciu do ktorej skupiny dané zariadenie patrí. Tabuľka môže byť kompletne uložená v ZC, alebo čiastočne v ostatných uzloch.
- Management - správa a údržba siete: Jedná sa o doplnkovú službu, ktorá sa stará o získanie informácií s okolitých blízkych sietí, spravuje vysielací výkon a iné.

ZCL je skupina clustrov a príkazov, ktoré sa používajú hlavne vo verejných profiloch podporované ZigBee Alianciou. Niektoré z nich slúžia na rozšírenie funkcionality ZigBee špecifikácie. ZCL sú organizované do domén ako napríklad osvetlenie, či vykurovanie. Pomocou týchto domén a tiež mnoho iných, definuje akési všeobecné pravidlá pre použitie. V prípade svetla to môže byť zapnúť/vypnúť osvetlenie. Na druhej strane však môžu existovať verejné profily, ktoré si definujú svoje vlastné clustre. Používanie ZCL nie je povinné. Vývojár aplikácie si môže vytvoriť svoje vlastné clustre.

## 2 SPÔSOBY PÁROVANIA ZARIADENÍ V ZIGBEE SIEŤACH(COMMISSIONING)

Párovanie je proces prepájania ZigBee aplikácii. Je tým myslené, aby napríklad vypínač zo svetla neovládal kotol a podobne. Proces párovania zariadení je na prvý pohľad triviálna záležitosť, ktorá však pri dôkladnom skúmaní nesie v sebe veľa prekážok a úvah. Načrtnem aspoň zopár situácií, ktoré sa v tomto procese riešia:

- Vyhľadanie vhodnej siete, do ktorej sa máme pripojiť
- Pripojenie sa do správnej siete
- Získanie zabezpečovacieho kľúča
- Rozhodnúť, s ktorými uzlami budeme komunikovať
- Akým spôsobom budeme v danej sieti komunikovať, či už skupinovo, priamo ...
- Stanoviť, čo spraviť v prípade, že nastala porucha na niektorom zariadení (výmena batérií a iné).

ZigBee poskytuje sadu niekoľkých primitív na vyriešenie spomínaných aspektov pri párovaní. Rozlišujeme tri odlišné scenáre, ktoré pokrývajú väčšinu problémov párovania (strana 333, [10]).

- Jednoduché párovanie - simple commissioning
- Metóda motýľ párovania - butterfly commissioning
- Užívateľské párovanie - custom commissioning

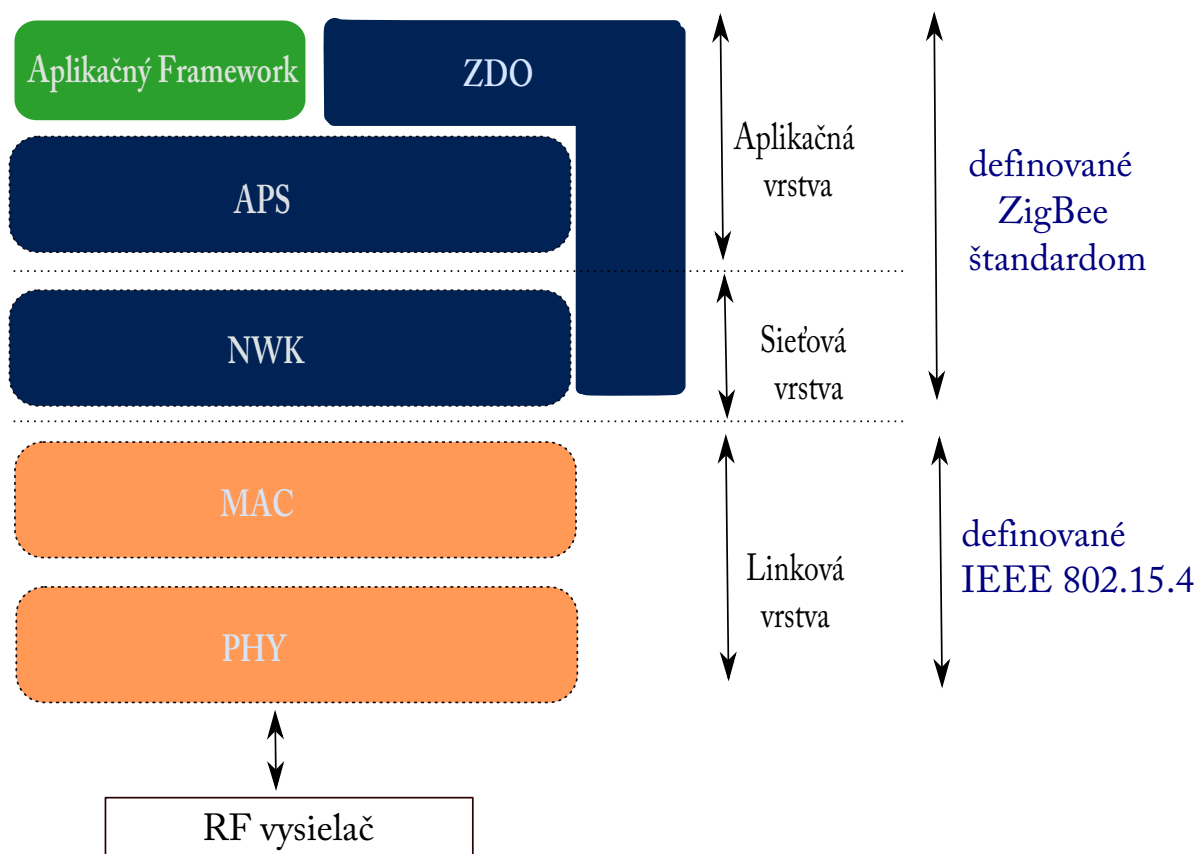
Prvé dve spomínané možnosti párovania uvažujú možnosť spolupracovania zariadení od nezávislých výrobcov a bez ich znalostí o prvom pripojení sa do siete. Posledná možnosť už predpokladá aspoň čiastočnú znalosť siete, do ktorej sa bude zariadenie pripájať, prípadne aj informácie o uzloch, s ktorými bude komunikovať. ZigBee využíva hlavne "Motýlí model" párovania. Idea spočíva v tom, že pri "zrodení" zariadenia ľubovoľná informácia môže byť nakonfigurovaná do zariadenia statickým spôsobom pri výrobe. Bez znalosti siete, do ktorej sa bude pripájať a bez dodatočnej konfigurácie by malo byť schopné sa po niekoľkých krokoch pripojiť a stabilne pracovať. Na tomto modeli som postavil aj svoju prácu.

ZigBee tvorí niekoľko objektov aplikačnej vrstvy. S ich využitím môžeme prevádzať rozličné úlohy, ku ktorým patrí i párovanie. Zamerajme sa na tieto :

- ZDO Obsahuje funkcie pre hľadanie a a pripojenie sa do siete. Využíva pri tom i časti sieťovej vrstvy. Pracuje vo vnútri každého zariadenia. Rozhodnutie, do ktorej siete sa zariadenie pripojí je vyhodnocované na základe parametrov získaných z jednotlivých vrstiev.
- ZDP - obsahuje zariadenie (uzol) a obsluhu (aplikáciu) objavovania sietí.

- ZCL - poskytuje bezdrôtové logické zoskupovanie zariadení a menežment siete.
- The Commissioning Cluster - poskytuje štandardné bezdrôtové prostriedky pre nastavovanie bezpečnostných kľúčov, PAN ID, masiek jednotlivých kanálov a správu adries.

Hierarchiu vrstiev v ZigBee môžeme vidieť na obrázku č. 2.1.

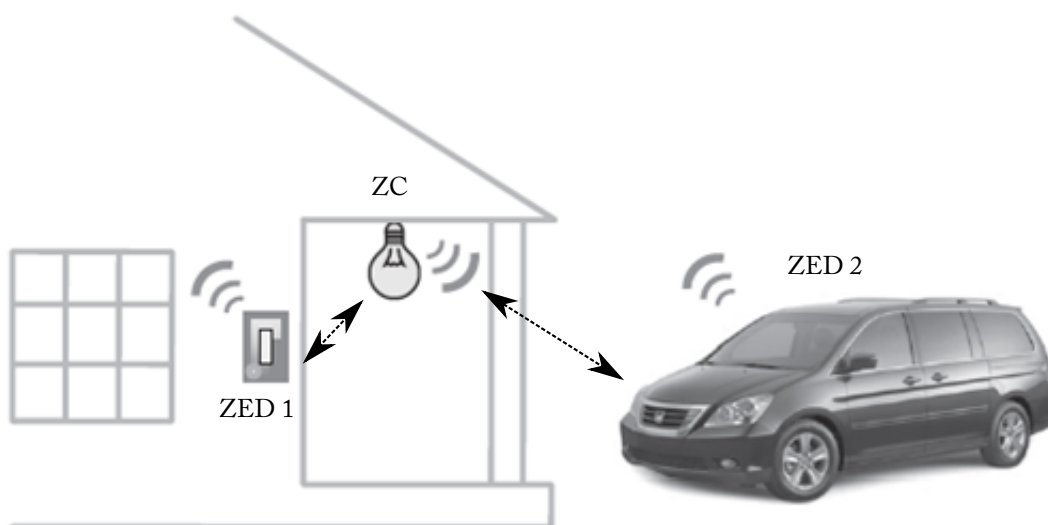


Obr. 2.1: Architektúra ZigBee/IEEE 802.15.4 vrstiev a objektov

## 2.1 Jednoduché Párovanie

Táto technika je implementovaná v ZigBee HA profile a je určená pre domácich užívateľov a inštalátorov bez nejakej hlbšej znalosti "počítačového" sveta. Ako príklad využijeme zapínanie a vypínanie vonkajšieho osvetlenia. Osvetlenie zapíname buď stlačením vypínača, alebo pomocou diaľkového ovládania. Typ párovania s topológiou hviezda je znázornený na obrázku č.2.2.

V tomto prípade vypínač sa chová ako ZED a svetlo ako ZC. Oba pri počiatocnom spúšťaní siete sa musia spárovať. HA aplikačný profil špecifikuje párovacie parametre pre toto jednoduché spustenie. Princíp spárovania zariadení je nasledovný:



Obr. 2.2: ZigBee jednoduché párovanie

1. Každý uzol pri prvom spustení nevie nič o nikom ani o sebe, má iba svoju MAC adresu a aplikačný profil. Nemá tiež žiadne informácie o PAN ID (na začiatku je nastavené na 0xffff) a nemá rozšírené PAN ID (nastavené na 0x0000000000), nie je mu priradený žiadny komunikačný kanál (nastavené na 0x07fff800), nemá žiaden sieťový kľúč a nemá žiadne profilové ID (bude sa pripájať na stack buď na 0x01 alebo 0x02).
2. Po zapnutí napájania uzol začne skenovať sieť. Hneď ako nájde sieť, ktorá má povolené *povolenie pripojenia* pripojí sa do tejto siete a získa od príslušného ZC sieťovú adresu a sieťový kľúč. Jediné zariadenie podľa tohto obrázku, ktoré je schopné formovať sieť je teda ZC.
3. Po pridelení potrebných parametrov na rozpoznanie zariadenia ZC vypne mód *povolenie pripojenia* na dve minúty potom, ako obdržal požiadavku od žiadostí o pripojenie od iného zariadenia. Spôsob ako vyvolať požiadavku o pripojenie k sieti je tiež možné sformovať manuálnym stlačením tlačítka na danom zariadení.
4. V poslednej fáze si ZC ukladá do NVM údaje o pripojenom zariadení, takže v tomto okamihu môže prísť o napájanie a pritom nepríde o dáta.
5. ZC a ZED opakujú proces párovania dovtedy, kým sa nepripoja.

Problém v týchto sieťach nastáva, keď by sme mali v dosahu iné siete. Vtedy by sa naše zariadenia nevedeli spárovať, pretože by nevedeli ku komu sa vlastne majú pripojiť. Riešenie v tomto okamihu spočíva vo vypnutí "cudzích" zariadení a nechať spárovať sa zariadenia tie, ktoré formujú novú sieť. Medzi výhody patrí rýchle

a jednoduché uvedenie zariadení do prevádzky.

## 2.2 Komerčné párovanie

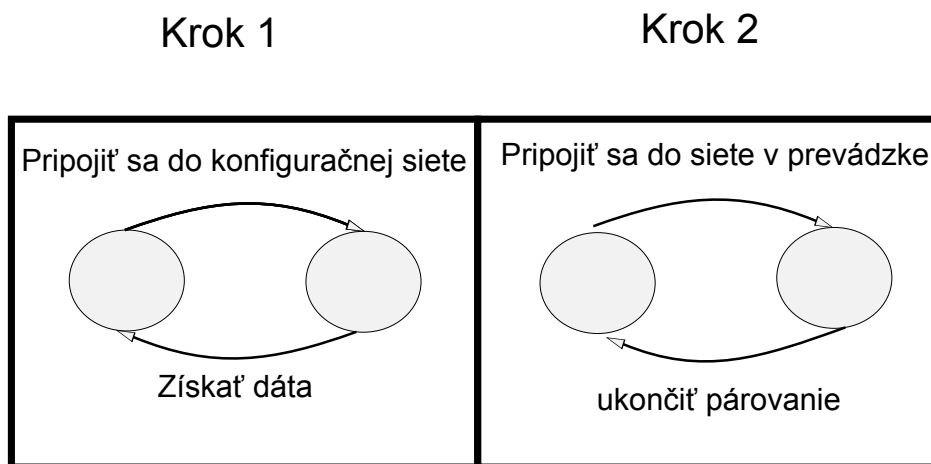
Pre jednoduché siete pozostávajúce z malého počtu uzlov je jednoduché párovanie dostačujúce. Pre rozsiahlejšie siete, alebo veľké skupiny tvorené niekoľkými sieťami, aké môžeme nájsť napríklad v hoteloch, zdravotníckych budovách či komerčných budovách je spôsob jednoduchého párovania nedostatočný. Požiadavky na tieto siete sú omnoho väčšie ako pri jednoduchých sieťach. V týchto sieťach sa stretávame s prepracovaneším prístupom, či už z hľadiska plánovania, alebo zabezpečenia. Inštalčné procesy sietí, prevádzka a údržba, to sú všetko dôležité aspekty pre plánovacie požiadavky. Pre aplikácie tohto typu musia byť siete plánované "predčasovo" a spôsobovať minimálne, alebo žiadne výpadky. Technici využívajú pôdorysy budov kvôli prehľadnému rozloženiu uzlov. Testujú, či pokrytie signálu medzi zariadeniami je dostatočné pre kvalitný prenos a či inštalované zariadenia (chladiace jednotky, termostaty, svetlá, zamykacie dvere atď ) pracujú tak ako sa očakáva. Pri údržbe a spravovaní sietí sa využíva inštalčných zariadení, pomocou ktorých sa testuje správnosť fungovania zariadenia v sieti a jeho oprava v prípade nespávnej funkcionality, alebo po prípade jeho náhrada pri zlyhaní v sieti. Často môžeme v tomto type sietí vidieť zariadenia od niekoľkých výrobcov, ako napríklad vypínače od firmy Siemens, vykurovací systém od Honeywell, klimatizácia od Schneider Electric, svetlá od firmy Phillips atď (strana 345,[10]). Zariadenia sú od výroby naprogramované a nastavené tak, aby ihneď po uvedení do prevádzky sa automaticky pripojili do existujúcej siete na rozšírenej PAN ID 0x00f0c27710000000. ZigBee definuje túto rozšírenú PAN ID, takže zariadenia od niekoľkých výrobcov môžu byť párované touto cestou.

Princíp spočíva v dvoch krokoch:

- pripojiť sa do existujúcej siete a prijať dáta s nastaveniami siete pre vlastný chod zariadenia.
- Reset zariadenia a pripojenie sa do tejto siete už s nastavenými parametrami.

Proces komerčného párovania je znázornený na obrázku č. 2.3. Zariadenie získa rozšírenú PAN ID adresu (z dôvodu výskytu viacerých zariadení odlišného výrobcu) z konfiguračného nástroja a potom sa po resete cez Commissioning Cluster pripojí do siete s PAN ID adresou tohto nástroja. Dokončí proces párovania a začína samostatne pracovať v sieti.

Výhody komerčného párovania spočívajú v detailnom plánovaní siete, od fyzického rozloženia, ktoré je zakreslené v plánoch budov, či objektov, až po logické spojenie uzlov spolu s ich parametrami. Veľmi často sa využíva externého konfiguračného zariadenia. Nevýhody: vyžaduje odborníkov pre konfigurovanie siete. Pri



Obr. 2.3: Komerčné párovanie s dvoma krokmi

výskyte problému krátkodobého výpadku zariadenia, vyžadujú prítomnosť človeka na obnovenie chodu siete.

## 2.3 Uživateľské párovanie

Spojazzdnenie siete je okamžité po vybalení zariadení z krabice. Aj takto sa dá charakterizovať prístup uživateľského párovania. Zariadenia totižto po zapnutí napájania sami formujú unikátnu sieť s unikátnym PAN a všetko je zrazu samo napojené a komunikuje. Oblasť využitia je najmä v pohybujúcich sa objektoch ako napríklad preprava surovín pomocou nákladného automobilu. Majme Tovar, ktorý počas prepravy musí byť neustále chladený na určitú teplotu, s určitou vlhkosťou .... Vďaka sformovaniu ZigBee siete vo vnútri chladiaceho boxu sme schopný nielen splniť tieto požiadavky, ale taktiež zaznamenávať ich parametre počas prepravy a následne vyhodnocovať v centrálnom systéme niekde v skladoch.

Ako sa prenášajú dáta z jednej siete do druhej? Sformulujme niekoľko predpokladov na prenos dát medzi týmito sieťami :

- Brána (*Gateway*), kam sú dáta sťahované, tvorí uzol 0x0000
- Senzory počas prepravy sú v móde zberania dát v určitom časovom intervale
- Raz za tri minúty senzory vyšlú *beacon request* pre hľadanie nových sietí
- Za potenciálne siete sú považované tie, ktoré sa nachádzajú v určitom rozsahu rozšírených PAN ID
- Pokiaľ sa nájde rovnaká sieť, z ktorej sa zariadenie už odpojilo, tak sa opätovne nepripája



- Pokiaľ sa našla nová sieť vyhovujúca predošlým požiadavkám zariadenia, sú spojené s touto sieťou a nastáva prenos dát

Všimnime si, že v sieti nedochádza k žiadnemu párovaniu. Zariadenia presne vedia s kým komunikujú, a ktoré zariadenie má akú úlohu. Tento mechanizmus je naprogramovaný už vo výrobe. Tento typ párovania je vhodný len pre konkrétny typ aplikácie. Je náročný na nastavenie iného typu prevádzky v porovnaní s komerčným párovaním.

## 2.4 Vyhodnotenie spôsobov párovania a výber typu pre aplikáciu

Scenária popisované v predchádzajúcich bodoch možno zhrnúť do 3 módov párovania. Ich rozdelenie do jednotlivých kategórií závisí na úrovni kontroly zariadení pri vzájomnom párovaní a na poskytnutom komforte ovládania pre užívateľa.

ZigBee Aliancia definuje tri *módy párovania*<sup>1</sup>, popisujúce tri rozličné prípady, ktoré môžu nastať pri párovaní:

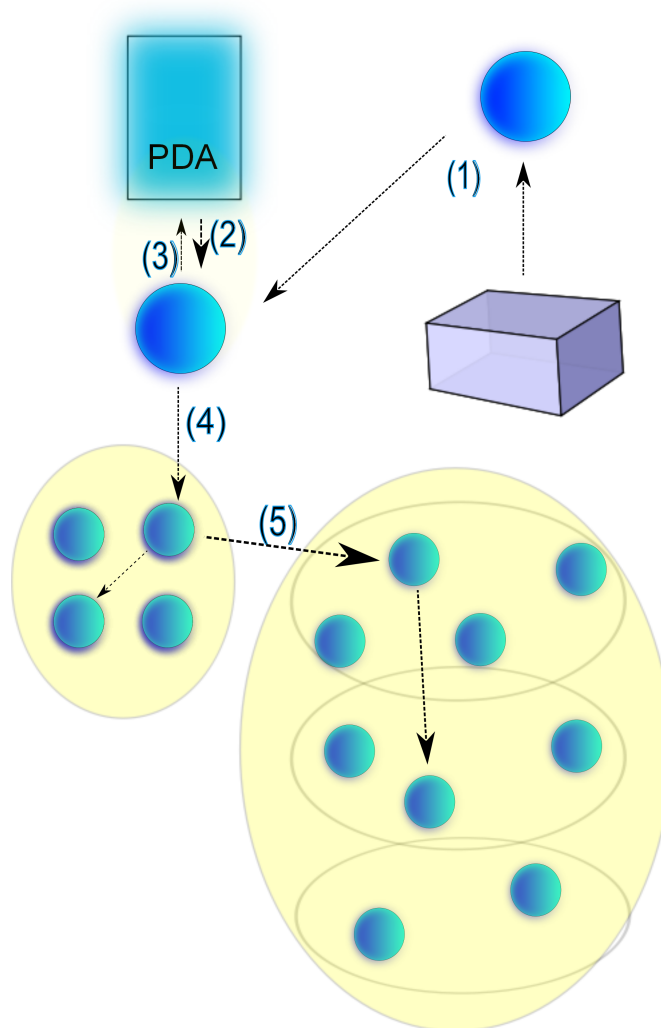
- **A - mód(Atomické):** V tomto móde sú zariadenia a siete schopné párovania sa bez ľudského zásahu.
- **E - mód(Jednoduché):** Uvedenie zariadenia v tomto móde do prevádzky vyžaduje minimálny zásah užívateľa alebo inštalatéra siete.
- **S - mód(Systémové):** V systémovej móde sú viac využívané rôzne konfiguračné nástroje, ktoré umožňujú dôkladnejšiu analýzu siete, jej nastavovanie a spravovanie.

V praxi sa častokrát i u jednoduchých sietí stretávame s prípadmi, kedy sú využité kombinácie týchto módov. Veľké firmy ako Daintree[1], alebo Atalum[2], ktoré majú vedúce pozície v oblasti bezdrôtových sietí ZigBee poskytujú riešenia hlavne v prepracovaných konfiguračných nástrojoch, pomocou ktorých sa zariadenia pridávajú do siete. Príklad môžeme vidieť na obrázku č. 2.4. Postup pripájania sa zariadenia do siete je nasledovný:

1. nové zariadenie je na začiatku po spustení sa pridať do každej siete.
2. Technik využije konfiguračného nástroja pomocou ktorej vytvorí dočasnú sieť pre novo pripájajúce sa zariadenie.
3. zariadenie sa pripojí do tejto siete.
4. Pomocou konfiguračného nástroja nastaví parametre zariadeniu, aby sa mohlo pripojiť do prevádzkovej siete.

---

<sup>1</sup>Založené na Európskom KNX modele



Obr. 2.4: princíp procesu pripojenia sa do siete

5. ak je potrebné, zariadenie sa môže neskôr pripojiť do väčšej siete.

## 2.5 Vlastné Riešenie Aplikácie

Pripájanie zariadení do siete z hľadiska flexibility som zhrnul do tabuľky 2.1. Pri manuálnom type pripájania zariadení do siete sa využíva procesu *binding*. Funkcionalitu si môžeme predstaviť na nasledovnom príklade: ZC sformuje sieť. Zariadenie, ktoré sa chce k tejto sieti pripojiť musí vyslať signál ZC. Toto sa prevádza napríklad stlačením tlačítka. Keď ZC prijme *bind* signál od niekeho zariadenia, automaticky si ho pridá do svojej siete. Tento typ riešenia nie je vhodný pre väčšie siete jednak z hľadiska polohy zariadení (snímače môžu byť umiestnené vo veľkej výške a sú často

Typ párovania	Výhody	Nevýhody
<i>Manuálne</i>	jednoduché na programovanie a na inštaláciu. Technik nemusí byť odborník v danej oblasti, aby sám zostavil sieť.	V prítomnosti čo i len ďalšej siete dochádza k neurčitosti k spárovaniu. Zariadenia nevedia ktoré ku ktorému sa má priradiť a veľmi ľahko sa môže stať, že sused bude ovládať našu sieť, po prípade my jeho.
<i>Automatické</i>	možnosť párovania siete i za predpokladu prítomnosti iných sietí. Sieť sa javí ako tichá v tom zmysle, že nepotrebuje žiadny vonkajší zásah do konfigurácie, respektíve do formovania siete.	Náročnosť na kód, potreba na-konfigurovať sieť už vopred, aby koordinátor vedel komu má priradiť akú úlohu a kto s kým bude komunikovať.

Tab. 2.1: Vyhodnotenie spôsobov párovania

nedostupné) a tiež z dôvodu výskytu iných sietí v dosahu.

V práci sa zameriavam na problematiku automatického párovania. Využívam poznatky o komerčnom párovaní. Za hlavný cieľ som si stanovil minimálny vplyv užívateľa pri výskyte chyby, ako napríklad výpadok zariadenia pri energetickej absencii. V praxi to znamená, že užívateľ je potrebný pri konfigurácii siete, pri jej počiatočnom spustení a pri výmene zariadenia za nové. Pri výpadku (reštarte) niektorého z koncových zariadení sa koordinátor automaticky postará o jeho opätovné pripojenie, pokiaľ sa znovu vyskytne v sieti. Pre konfiguráciu siete bude využívať konfiguračný nástroj v grafickej podobe.

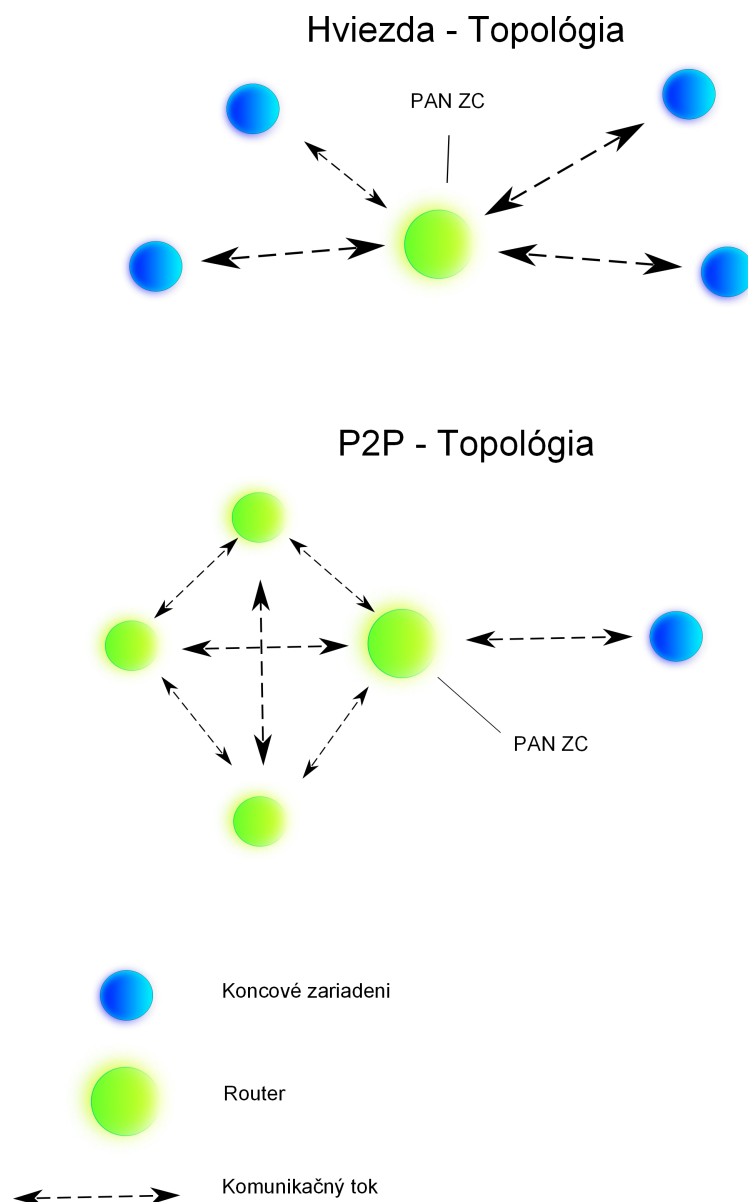
V samotnej práci je riešené:

- **automatické pripojenie uzla do siete a jeho konfigurácia:** zaradenie uzlu do siete v prípade, že už v sieti fungoval, teda bol zahrnutý do počiatočnej konfigurácie siete. V praxi to znamená, že po pripojení sa do tej istej siete, v ktorej už vykonával prevádzku, ZC automaticky po overení MAC adresy posielala konfiguračné údaje z uloženej tabuľky tohto uzla.
- **vytvorenie konfiguračného programu:** Konfiguračný program slúži na prvotné nastavenie uzlov v sieti. V tomto bode sa rieši aj komunikácia medzi ZC a PC, na ktorom je programová aplikácia.

### 3 VLASTNOSTI TESTOVANEJ SIETE

#### 3.1 Topológia Siete

Podľa špecifikácie IEEE 802.15.4 sformovaná sieť musí byť v jednom z dvoch typov topológií: hviezda(star), alebo P2P.[11] Topológie sú znázornené na obrázku č. 3.1.



Obr. 3.1: Príklad topológie typu hviezda a P2P

Topológia typu hviezda sa vyznačuje usporiadaním jednotlivých zariadení tak, že sieť tvorí jedno centrálné zariadenie označované ako PAN ZC a všetky ostatné zariadenia komunikujú priamo s ním [22]. Zariadenia majú obvyčajne pridelený nejaký

typ úlohy s tým, že jedno zariadenie slúži na inicializáciu a iné zas na koncový prístup. PAN ZC môže mať pridelenú tiež nejakú špecifickú aplikáciu, ale na rozdiel od iných zariadení, môže využívať vlastnosti všetkých zariadení v sieti a to jak inicializáciu siete, koncový prístup, alebo len smerovanie dát. PAN ZC je primárne zariadenie v sieti. Všetky zariadenia pracujúce v sieti sa vyznačujú unikátnym typom 64-bitovej MAC adresy. MAC adresy sa používajú na priame adresovanie vo vnútri PAN. Po asociácii zariadenia do siete, ktorú vytvoril PAN ZC, je automaticky vystavená skrátaná adresa(short address) pre pripojené zariadenie, a pre ďalšiu komunikáciu s týmto zariadením je použitá práve táto adresa. PAN ZC býva často napájaný stálym zdrojom elektrickej energie, kým pre ostatné zariadenia slúži ako napájací zdroj batéria. Aplikácie, ktoré ťažia z výhod tejto topológie sú napríklad domáca automatizácia, periférie osobných počítačov, hračky, hry a osobná zdravotná starostlivosť [9].

Medzi ZigBee najviac používané sieťové topológie okrem hviezda(star) patria tiež spleť(mesh) a strom(tree). Každá z nich obsahuje radu výhod i nevýhod. V mojej práci využívam topológiu hviezda. Pri výbere topológie som zvažoval nasledovné kritéria:

- nie som limitovaný priestorovým usporiadaním zariadení(každé zariadenie má dosah na koordinátora).
- predpokladám využitie hlavne v oblasti domácej automatizácie.
- Koordinátor bude pripojený k trvalému zdroju napájania z dôsledku častej komunikácie so zariadeniami, alebo s PC, či PDA.

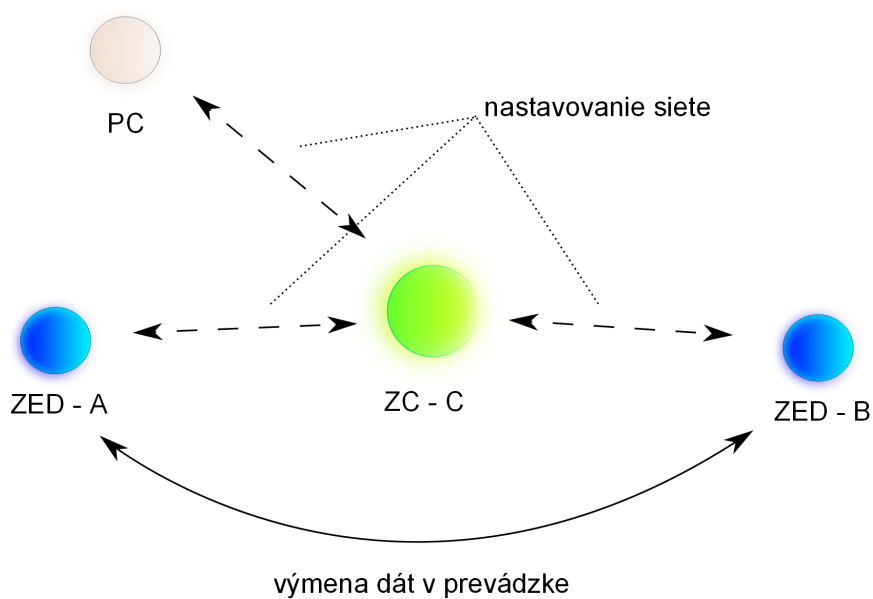
## 3.2 Formovanie Siete Star(Hviezda)

O formovanie siete sa stará sieťová vrstva. Základná štruktúra siete je na obrázku č. 3.1. Pri formovaní siete využívam nasledovný postup:

1. FFD zariadenie po spustení automaticky formuje sieť a stane sa PAN ZC. Všetky topológie typu hviezda pracujú nezávisle na ostatných sieťach typu hviezda v súčasnej prevádzke. Toto je dosiahnuté výberom unikátneho PAN identifikátora, ktorý nie je v dosahu žiadnej siete rovnaký. Keď koordinátor vyberie vhodný PAN identifikátor, automaticky dovoľuje ostatným zariadeniam sa pripojiť do tejto siete.
2. Spustíme ostatné zariadenia v sieti. Každé zo zariadení ihneď po pripojení sa do siete pošle svoje identifikačné údaje koordinátorovi.
3. Koordinátor vytvára a udržiava tabuľku zariadení v sieti a po vyžiadaní tejto tabuľky od externého zariadenia mu ju odošle.

4. Pomocou konfiguračného nástroja v externom zariadení vytvorí užívateľ pravidlá pre komunikáciu jednotlivých uzlov v sieti a odošle ju ZC.
5. ZC rozošle pomocou *Unicast správ* už správy konkrétnym uzlom. Tieto uzly už následne po prijatí konfiguračnej správy od ZC vedia, komu majú posielat' dáta.
6. Sieť sa prepne do stavu prevádzky. ZC udržiava chod siete.

Na obrázku č. 3.2 je znázornená výmena dát medzi konkrétnymi uzlami. Uzol A posielá dáta uzla B a naopak.



Obr. 3.2: Dátový tok v sieti

## 4 NÁVRH APLIKÁCIE

V návrhu aplikácie najskôr všeobecne popíšem význam a možnosti vytvorenej aplikácie, a v jednotlivých podkapitolách sa zameriam na detailný popis a etapy návrhu aplikácie. Kapitola *Návrh Aplikácie* je rozdelená na 3 podkapitoly :

1. návrh firmware pre jednotlivé uzle siete.
2. návrh grafickej aplikácie - konfiguračného nástroja bežiaceho na externom zariadení(PC, alebo PDA) a komunikácia medzi ZC a externým zariadením.
3. správa siete z pohľadu užívateľa.

Hlavným prínosom aplikácie je minimalizácia doby nepripojeného uzlu a pohodlnejšia konfigurácia zariadení.

- **Minimalizácia doby nepripojeného uzlu:** Zariadenie stratí na určitú dobu, prípadne natrvalo napájanie. Po obnovení sa zariadenie pripojí do siete a ZC mu automaticky posiela údaje pre komunikáciu v sieti. Toto platí za predpokladu, že zariadenie už v sieti pracovalo.
- **Pohodlnejšia konfigurácia zariadení:** Konfiguračný nástroj slúži na pripojenie sa ku koordinátorovi siete, získaní informácii o sieti - koľko sa tam nachádza uzlov a s akými nastaveniami, a na nastavenie dátového toku zariadení v prevádzke.

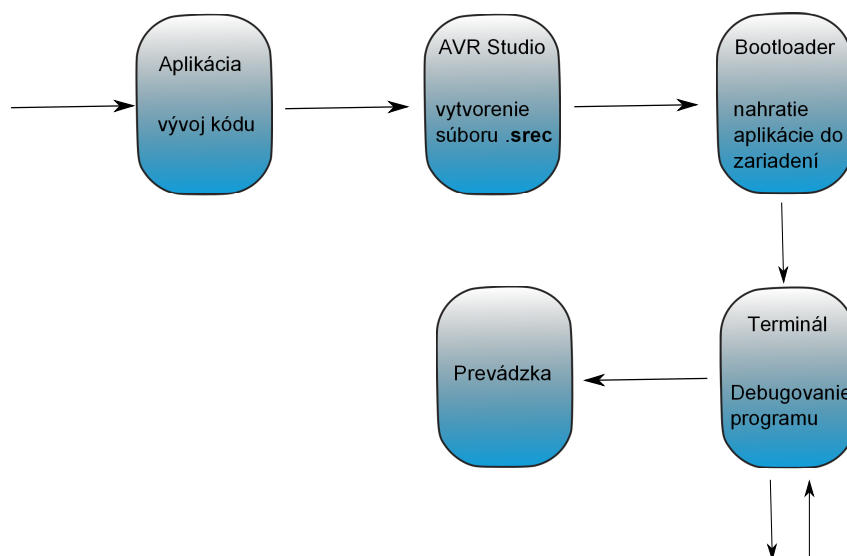
### 4.1 Firmware pre Jednotlivé Uzle Siete

Algoritmy pre jednotlivé uzly sú programované pod operačným systémom Windows 7 s inštalovaným vývojovým prostredím AVR Studio 4. Algoritmus je písaný v programovacím jazyku C.

Vývojové prostredie IDE(Integrované vývojové prostredie - Integrated Development Environment) je určené okrem iného i pre 8 bitové AVR aplikácie. IDE podporuje všetky nástroje od Atmelu, ktoré podporujú 8-bitové AVR architektúry. AVR Studio 4 poskytuje debugovaciu podporu pre písaný C kód - debugger; registre procesoru, pamäť a pohľad na vstupy/výstupy; plne programovaciu podporu.[3]

Spôsob nahrávania a debugovania algoritmu je znázornený na obrázku č.4.1. Vytvorený kód v jazyku C sa preloží a zkompiluje. Následne sa prilinkujú dôležité knižnice a ďalšie súbory do projektu, a vytvára sa finálny súbor **.srec**. Tento súbor je už tzv. Firmware pre jednotlivé uzly. Firmware sa nahrá do kontroléru pomocou Jtag alebo cez Bootloader.

Vyvíjané algoritmy som testoval na vývojových doskách od firmy MeshMetics s označením ZigBit 900[15]. ZigBit 900 obsahuje kompletný dizajn pre RF/MCU so



Obr. 4.1: Vývoj firmware do uzlov

všetkými potrebnými pasívnymi komponentmi.

Jednotlivé vývojové dosky MeshBean 900 pracujú na frekvencii 868/915MHZ a vyhovujú 802.15.4/ZigBee štandardu. Zahŕňajú v sebe teplotný senzor, senzor svetla a ďalšie periférne zariadenia. Jednotlivé vývojové dosky pracujú v sieti samostatne - obsahujú tak ako vysielač tak i prijímač jednotku na spracovanie signálu spolu s procesorom a ostatnými perifériami a tvoria tak bezdrôtové uzly schopné funkcie koordinátora, smerovača, alebo koncového zariadenia. Napájané sú pomocou USB, alebo dvoch AA batérii. Obsahujú externú RP-SMA anténu, tri LED, dve programovateľné tlačítka a jedno resetovacie tlačítko. Vývojová sada ZigBit 900 je predstavená na obrázku č. 4.2. Hlavný dôvod používania týchto vývojových setov je ich podpora pre *BitCloud ZigBee stack* API, ktoré značne uľahčujú prácu. Pomocou tohto API sa môžeme plne zamerať na písanie programu na vyššej úrovni. Pre obsluhu požiadavkou nižších úrovní mi stačí používať funkcie a rutiny z BitCloud API stacku [7].

**Bitcloud** je embedded software stack od firmy Atmel. Stack ponúka vývojovú firmware platformu pre spoľahlivé, rozšíriteľné a zabezpečené bezdrôtové aplikácie, bežiacie na hardwarových setoch od firmy Atmel. Bitcloud je dizajnovaný pre poskytnutie širokej škály možností pre užívateľa, s možnosťou uspokojovania aplikácií tak ako potrebuje. BitCloud je primárne určený pre aplikácie typu domáca automatizácia, komerčná automatizácia budov, automatizované meranie, sledovanie majetkov a priemyselná automatizácia.





Obr. 4.2: Vývojový set ZigBit 900[15]

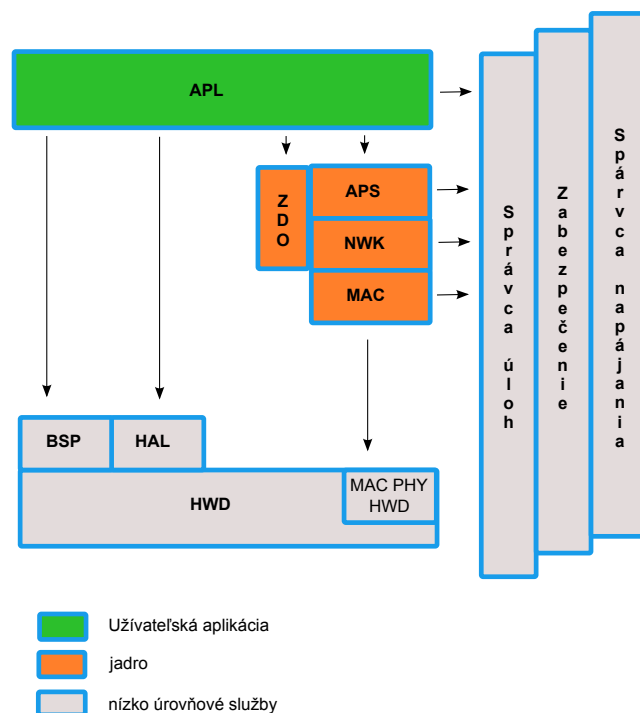
Bitcloud plne vyhovuje pre aplikácie bezdrôtového snímania a riadenia pod záštitou ZigBee štandardu IEEE 802.15.4. Poskytuje rozšíriteľné API, ktoré je v súlade so štandardom. API a jeho náležitosti sú navrhnuté tak, aby sa programátor nemusel starať o každý detail na hardwarovej úrovni. K tomu slúžia funkcie a rutiny, ktoré už len využíva pri písaní vlastného algoritmu.

**Architektúra Bitcloudu** je zobrazená na obrázku č. 4.3.

Vnútoraná architektúra Bitcloud rozdeľuje sieťový zásobník do logických vrstiev, tak ako to hovorí štandard IEEE 802.15.4 a ZigBee. Okrem jadra, ktoré obsahuje implementáciu protokolu, Bitcloud zahŕňa realizáciu :

- dodatočných vrstiev implementujúcich zdieľané služby ako napríklad správca úloh, zabezpečovacia vrstva a správca napájania.
- abstrakciu hardverovej vrstvy - HAL Hardware Abstraction Layer
- BSP Board Support Package

Spomenuté doplnky a mnohé iné okrem API umožňujú výrazne znížiť komplexitu aplikácie a tiež poskytujú jednoduchosť integrácie. BitCloud manuál poskytuje podrobnejšie informácie o jednotlivých API.[7]



Obr. 4.3: Architektúra BitCloud Software stacku

APS sa nachádza na najvyššej úrovni v jadre. Poskytuje najvyššiu úroveň sieťových komponentov spojenými s API, ktoré sú viditeľné v aplikácii. ZDO poskytuje hlavné služby pri formovaní siete, ako napríklad štart, reset, formovanie siete, pripájanie sa do siete ... ZDO taktiež definuje rôzne typy ZDP a funkcií (*objavenie zariadenia* ...).

Služby ako správca úloh, zabezpečenie a správa napájania sú dostupné užívateľovi v aplikácii a môžu byť modifikované.

**Správca úloh** je zásobníkový plánovač, ktorý sprostredkováva používanie MCU medzi vnútornými procesmi zásobníku a aplikáciou užívateľa. Správca úloh implementuje prioritný kooperatívny plánovač, ktorý je špeciálne navrhnutý pre viacvrstvový zásobník a vyhovuje požiadavkom pre časovo kritické sieťové protokoly.

Rutiny správcu napájania sú zodpovedné za korektné vypnutie všetkých komponentov zásobníka, uloženie stavu systému pri prechode do "spiacieho" stavu a obnovenie pôvodného stavu pri "zobudení" sa zariadenie.

HAL Hardware Abstraction Layer zahŕňa kompletný set API pre používanie hardverových zdrojov (watchdog časovače, EEPROM ...). Tiež zahŕňa rutiny pre ovládanie externých zariadení ako napríklad: IRQ, SPI, USART,  $I^2C$  a iné.

BSP Board Support Package zahŕňa kompletnú sadu ovládačov pre správu štandardných periférií (senzory, UID čip, tlačítka), umiestnené na vývojovom sete.

### 4.1.1 Návrh Algoritmu

Predpokladajme následovné stavy, ktoré sa môžu vyskytnúť pri prevádzkovaní siete:

1. *Výpadok zariadenia zo siete:* príčina môže byť krátkodobý výpadok napájania (vybitie batérie) a jeho opätovné zaradenie sa do siete. V práci beriem v úvahu výpadok len niektorých koncových zariadení ako sú smerovač a koncové zariadenia. Výpadok koordinátora by znamenalo nastavovanie konfigurácie celej siete od začiatku.
2. *Pridanie nového zariadenia do siete:* ako názov hovorí, užívateľ môže pridať do siete nové zariadenia. Počet zariadení je limitovaný veľkosťou pamäte v ZC. V ZC sa ukladajú informácie zo siete vo forme tabuľky. Preto i množstvo týchto tabuliek je limitované veľkosťou pamäte.
3. *Výmena zariadenia za nové:* Pri pokazení zariadenia je potreba ho nahradiť úplne novým zariadením. To znamená, že pre pridanie tohto zariadenia ZC musí vedieť, že toto zariadenie bude preberať úlohu pokazeného zariadenia a teda mu musí poslať údaje z uloženej tabuľky, odpovedajúce údajom pokazeného zariadenia. Nové zariadenie tak po pridelení do siete preberie úlohu starého zariadenia. Na tento úkon sa vyžaduje pripojenie sa do siete pomocou konfiguračného nástroja, pomocou ktorého nastavíme parametre pre nové zariadenie.
4. *Možnosť konfigurácie siete a nastavovanie parametrov:* úloha vyžaduje vytvorenie konfiguračného nástroja, pomocou ktorého užívateľ bude môcť spravovať sieť.

Výsledok práce pozostáva z troch algoritmov, samostatne bežiacich na zariadeniach zobrazených na obrázku č. 4.4.

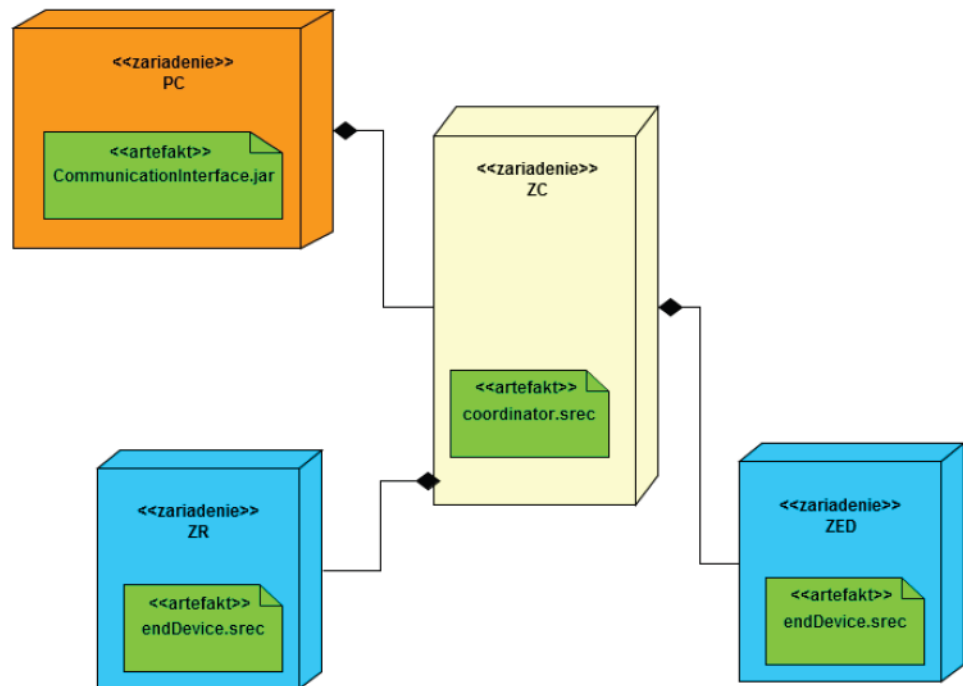
Ako možno vidieť na externom zariadení - PC beží program - CommunicationInterface.jar napísaný v programovacom jazyku Java<sup>1</sup>. V ZC je program napísaný v jazyku ANSI C a preložený do binárneho súboru *coordinator.srec*. V ZED a ZR je spoločný program v jazyku ANSI C preložený do binárneho súboru *endDevice.srec*.

**Príprava prostredia a adresárová štruktúra projektu:** Z domovských

stránok Atmel [3] som stiahol BitCloud Stack verziu 1.10. Adresárová štruktúra obsahuje okrem implementačnej časti BitCloudu aj časť dokumentačnú, ktorá popisuje už vytvorené funkcie a štruktúry, a časť pre vlastnú aplikáciu. V tejto časti som si vytvoril adresár s názvom pre daný uzol (Coordinator a EndDevice). Dôležité adresáre a súbory pre AVR Studio verziu 4.18 sú znázornené na obrázku č. 4.5.

---

<sup>1</sup>Java je programovací jazyk a počítačová platforma uvedená v roku 1995 firmou Sun Microsystems [16]



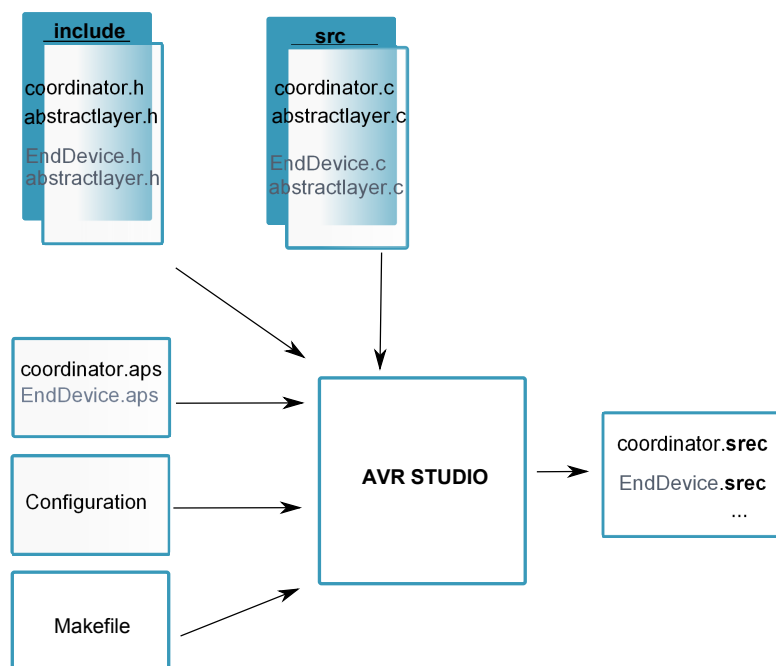
Obr. 4.4: Deployment diagram algoritmov bežiacich na rôznych zariadeniach

#### 4.1.2 Implementácia Algoritmu pre Koordinátor

Po nahratí zdrojového programu - firmware do zariadenia ZC, sa začne vykonávať nasledovný sled programu:

- inicializácia periférie a zariadenia
- sformovanie siete a registrovanie pripojených zariadení
- konfigurácia siete
- prevádzka siete

Hlavná funkcionálnosť je riešená formou stavového automatu, v ktorom sú logicky rozdelené sekcie kódu presne tak ako musia na seba nadväzovať v danom čase. V mnohých prípadoch je závislosť fungovania jednej funkcie závislá na inej a podobne. Preto sa využíva tzv. *CallBack funkcie*, ktoré potvrdzujú správnosť predaných, či doručených dát. Funkcie spodných vrstiev, ktoré slúžia či už na prácu s externými perifériami (IRQ, I<sup>2</sup>C, SPI, UART, 1-Wire), štandardnými perifériami (senzory,



Obr. 4.5: Hlavné súbory potrebné pre preklad projektu a výsledok prekladu

UID čipom, prepínačmi, tlačítkami), abstrakciu hardverovej vrstvy a podobne sú už implementované v BitCloud Stack [7].

Okrem samotného pripojenia sa do siete a prevádzky v nej, je paralelne po inicializácii periférií riešená komunikácia s externým zariadením, ktoré po pripojení je určené na konfiguráciu siete.

V komunikácii medzi zariadeniami ZC a ZED sa využíva dvoch komunikácií na výmenu dát. Jedná sa o :

1. Procesná komunikácia
2. Servisná komunikácia

Procesná komunikácia znamená výmenu dát už v danej aplikácii medzi zariadeniami. Príkladom môže byť nameranie teploty na zariadení ZED A a poslanie hodnôt zariadeniu ZED B. Tieto zariadenia už teda vedia, alebo majú informáciu o tom, že majú spolu komunikovať a tiež že ktorým smerom budú dáta posielat' (z A do B). O Servisnú komunikáciu sa stará koordinátor. Prebieha teda vždy medzi koordinátorom a jedným z koncových zariadení. Koordinátor komunikuje so zariadeniami vždy po jednom a správy si overuje či naozaj zariadeniam prišli. Tento typ komunikácie prebieha pri nastavovaní siete, zmene stavu siete.

## Popis Stavov v Stavovom Automate

Hlavným vláknom programu je funkcia *APL\_TaskHandler()*, v ktorej je stavový automat implementovaný. Zjednodušenú schému stavového automatu pre Koordinátor môžeme vidieť v prílohe A.1. Z neho vychádzajú nasledovné stavy programu:

### APP\_INITING\_STATE:

Tento stav je tiež nazývaný ako inicializačný stav a volá funkciu *initNetwork()*, pomocou ktorej sa inicializujú všetky periférie, alokuje sa miesto pre štruktúry, premenné. Ako prvé sa inicializujú periférne zariadenia, v prípade koordinátora sú to tlačítka a LED diódy, taktiež sériová komunikácia USART. Následne sa vyberá o aký typ zariadenia ide (ZC, alebo ZED, prípadne ZR) a táto informácia sa zapíše pomocou *ConfigServer Interface*<sup>2</sup> do zariadenia.

### APP\_NETWORK\_STARTING\_STATE:

Automat v tomto stave volá funkciu *startNetwork()*, v ktorej sa formuje sieť, registrujú sa zariadenia, vytvárajú sa komunikačné kanály a v ktorej koordinátor posielajú pravidlá komunikácie jednotlivým zariadeniam.

Pomocou zelenej LED, ktorá bliká v 0.5s intervale indikujem, že sa formuje sieť. Pri jej úspešnom sformovaní príde potvrdenie zo spodnej vrstvy a následne ako je sieť už vytvorená, LED svieti neustále. Pri vytváraní siete používam funkciu *ZDO\_StartNetworkReq()*.

Následuje registrácia endpointov a vytvorenie komunikačných kanálov. Tieto kanály sa vytvárajú hneď dva a to jeden pre procesnú komunikáciu a druhý pre servisnú komunikáciu. Spôsob vytvorenia a vlastnosti týchto kanálov sú definované pomocou *simpleDescriptoru*. *SimpleDescriptor* je súhrn pravidiel uložených v štruktúre pre daný endpoint. Každé zaregistrovanie zariadenia má za následok inkrementovanie vnútornej premennej *actual\_number\_of\_devices*. Vyčítaním jej hodnoty viem presne koľko zariadení sa už v sieti ku koordinátorovi pripojilo. Po dosiahnutí maximálneho počtu zariadení, ktoré je vnútorne definované vo firmveri ZC a je na začiatku nastavená užívateľom, prechádza zariadenie do ďalšieho stavu.

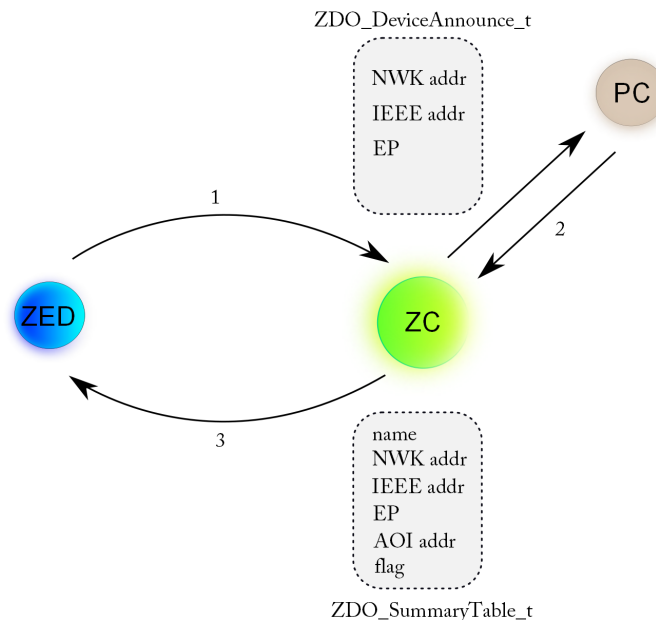
Po každom zaregistrovaní zariadenia je koordinátorovi posielaná štruktúra s informáciami o pripojenom uzle, ktorá obsahuje položky *ZDO\_DeviceAnnounce\_t* štruktúry. Zoznam položiek štruktúry a ich význam je nasledovný:

- name : v názve je uložené poradové číslo zariadenia. Jeho hodnota závisí, v akom čase, respektíve poradí bolo zariadenie pripojené do siete.

---

<sup>2</sup>BitCloud poskytuje sadu rozšírených konfiguračných parametrov, ktoré popisujú odlišné aspekty chovania sa uzlu v sieti. Príkladom je nastavenie typu zariadení, vysielač výkon a iné parametre

- nwkAddr: Sieťová adresa zariadenia
- ieeeAddr: Mac adresa zariadenia
- ep: Enpoint na ktorom prebieha procesná komunikácia



Obr. 4.6: proces konfigurácie siete

Tieto položky sa spolu s poradovým číslom zariadenia ukladajú do hlavnej tabuľky *ZDO\_SummaryTable\_t*, ktorá spolu ďalšími položkami tvorí pole štruktúr. Toto pole sa po naplnení a na vyžiadanie od užívateľa preniesie do PC. V počítači sa upravujú komunikačné parametre, nastaví sa smerovanie zariadení a odošle sa do späť do ZC. Novovzniknutá štruktúra vyzerá nasledovne:

- name
- nwkAddr
- ieeeAddr
- ep
- aoIAddr : adresa komu bude zariadenie posilať dáta v prevádzke
- SendReceiveFlag : rezervovaný flag pre ďalšie použitie

Ďalšou úlohou ZC v tomto stave je odoslanie komunikačných údajov jednotlivým uzlom. Odosielanie údajov sa deje pomocou funkcie *sendCommunicationData()*, ktorá ako adresu, na ktorú majú byť dáta doručené využíva presne parameter *nwkAddr* a je volaná presne toľkokrát koľko je nadefinovaný počet zariadení v sieti.

Toto všetko prebieha už v spomínanej Servisnej Komunikácii, na vyhradenom komunikačnom kanále (EndPoint = 4).

Detail konfigurácie siete je znázornený na obrázku č.4.6. V prvom kroku koncové zariadenie posiela svoje informácie koordinátorovi siete. V druhom kroku sú po pripojení počítača tieto informácie spracované a odoslané späť koordinátorovi. V treťom kroku po ich obdržaní koordinátor posiela komunikačné informácie danému uzlu. Koncové zariadenie nastaví predpísanú komunikáciu a začína prevádzku.

#### **APP\_IN\_NETWORK\_STATE:**

V stave APP\_IN\_NETWORK\_STATE je už sieť sformovaná a nastavená. Všetky zariadenia vedia s kým budú uskutočňovať výmenu dát. Zariadenia v tomto stave prechádzajú do stavu plnej prevádzky. Vďaka dvom rozdielnym komunikačným kanálom, môže aplikácia bežať samostatne bez zasahovania do servisnej prevádzky a naopak. V prípade, že sa vyskytne porucha na zariadení, prípadne iný typ zlyhania prevádzky zariadenia, je komunikácia automaticky presmerovaná cez servisný kanál a môže byť konfigurovaná práve cez neho.

V ZC je implementovaná časť pre opätovné prihlásenie sa zariadenia do siete toho istého zariadenia v prípade, že sa v sieti už nachádzal. Príkladom môže byť zlyhanie napájania v prípade, že sú vybité batérie. Po ich vymenení sa koncové zariadenie opäť pripojí do siete s tou istou mac ale rozdielnou sieťovou adresou. ZC po každom takomto pripojení sa zariadenia do siete kontroluje výskyt mac adresy s mac adresami uloženými v konfiguračnej tabuľke. Koordinátor totižto má už v tomto čase v sebe uložené ku akej mac adrese prislúcha aká sieťová adresa. Po jej prehľadaní a overení totožnosti ZC automaticky posiela do ZED sieťovú adresu presne takú, s akou bolo zariadenie pôvodne nakonfigurované pri prvom uvedení siete do prevádzky.

### **4.1.3 Implementácia Algoritmu pre Koncové Zariadenie a Smerovač**

Program pre koncové zariadenie predstavuje rozhodne jednoduchší algoritmus. Hlavnou podstatou je spracovanie požiadavky koordinátora siete a k danému požiadavku poslať korektnú odpoveď. Opäť ako u ZC tak aj v ZED, či ZR prebiehajú dve paralelné komunikácie - servisná a procesná. Stavový automat programu je znázornený v prílohe B.1.



## Popis Stavov v Stavovom Automate

Stavový automat je implementovaný v hlavnom vlákne *APL\_TaskHandler()*. Automat obsahuje 3 hlavné stavy:

- **APP\_INITING\_STATE**: inicializačný stav
- **APP\_NETWORK\_STARTING\_STATE**: stav pripojenia sa do už existujúcej siete
- **APP\_IN\_NETWORK\_STATE**: prevádzka v sieti, výmena procesných dát

### **APP\_INITING\_STATE**

Ako názov hovorí, v zariadení prebieha inicializácia periférií, alokácia pamäte a podobne. Volá sa funkcia *initNetwork()*. Proces nastavovania parametrov je obdobný ako u ZC. Po úspešnej inicializácii prechádza zariadenie do druhého stavu.

### **APP\_NETWORK\_STARTING\_STATE**

V stave formovania siete, zariadenie skenuje existujúce siete a pripája sa do správnej. Komunikácia prebieha na servisnej úrovni. Po registrácii komunikačných kanálov - *endpoint*, koncové zariadenie posiela svoje identifikačné údaje, ktorými sú : endpoint, sieťová adresa a mac adresa do ZC. V zápatí na to očakáva od neho konfiguračné dáta s informáciami, na akom komunikačnom kanáli bude komunikovať s ostatnými zariadeniami, a s akým zariadením si bude vymieňať dáta. Po ich prijatí nastaví ZED spomínané údaje a prepína sa do stavu plnej prevádzky.

### **APP\_IN\_NETWORK\_STATE**

V poslednom stave už je dokončená konfigurácia siete. Vďaka paralelnej komunikácii (komunikácia na dvoch rozdielnych komunikačných kanáloch) je možné tiež konfigurovať zariadenie v prevádzke. Taktiež je možné presmerovanie a výpis chybových stavov v sieti (výpadok, stratenie signálu ...) do terminálu. Zariadenie po každom novom prihlásení do siete, posiela ZC svoje identifikačné údaje prostredníctvom servisného kanálu, čo umožňuje jeho identifikáciu a možnosť opätovného zaradenia do siete bez zásahu užívateľa.

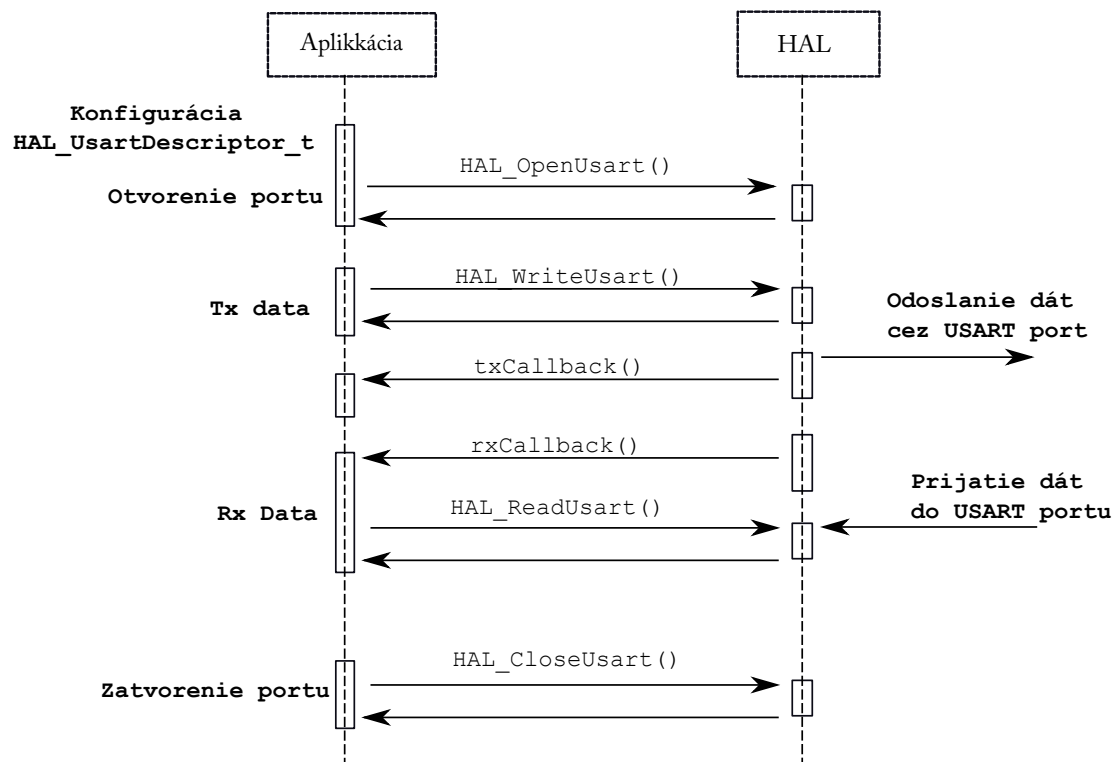
## 4.2 Návrh Softvéru

V tejto kapitole popisujem základnú štruktúru WSN softvéru s demonštračnou aplikáciou. Základ tvoria 4 zariadenia, jedno je konfigurované ako koordinátor siete, dve zariadenia preberajú úlohu koncových zariadení, a počítač, na ktorom je spustená aplikácia. Spôsob komunikácie sa riadi podľa obrázka č.4.4. Sieťová topológia je hviezda.

## 4.2.1 Nízkoúrovňová Komunikácia a Protokol

### USART

Na strane ZC som využil knižnic BitCloudu, ktorý poskytuje API pre USART rozhranie. Prvotné nastavenie komunikácie sa konfiguruje prostredníctvom *HAL\_UsartDescriptor\_t* štruktúry. V nej sa nastaví potrebné parametre pre komunikáciu ako prenosová rýchlosť, synchronný alebo asynchronný mód a iné (strana 35,[7]). Využívam tzv. *Callback* mód. Princíp je znázornený na obrázku č.4.7.



Obr. 4.7: USART výmena dát v callback móde

Parametre komunikácie majú nasledovné parametre:

- asynchronný mód
- prenosová rýchlosť: 38400 baudov
- počet bitov: 8
- parita: žiadna
- stop bit: 1

Prijímacie dáta z USART portu sú spracované v callback funkcií - *naplnBufferString()*. Pre prijatie správnych dát je potrebné ich odlíšiť od dát, ktoré sa vyskytujú v prenosovom kanále. Na to som zostavil rámec špecifického tvaru. Príklad rámcu pre jedno zariadenie má nasledovný tvar:

[[2|6|0 : 1 : 0 : 0 : 11 : 2B : F9 : 7C|2|1|1]]

Každý rámec začína "[" a končí "]". Každá špecifická položka v rámci je oddelená separátorom |. Maximálna veľkosť rámca pre jedno zariadenie je 59 znakov. Význam jednotlivých položiek z ľava je nasledovný:

1. číslo zariadenia: záleží na poradí v akom sa prihlási zariadenie do siete
2. sieťová adresa
3. MAC adresa
4. endpoint
5. cieľová adresa: je to adresa na ktorú bude zariadenie posielat' dáta
6. flag: rezervovaná položka

Na rozpoznanie správy zakódovanej v tomto rámci používam funkciu *receive()*, ktorá prijatý rámec rozparsuje do jednotlivých položiek.

### Obsluha rxCallback funkcie

Po prijatí dát na port USART sú v rxCallback funkcii presmerované dáta do funkcie *naplnBufferString()*. Vývojový diagram pre spracovanie dát vo vnútri funkcie je znázornený v prílohe č. D.1. Ako prvé sa porovnávajú prijaté dáta so znakom "[", ktoré sú prenášané na linke. Tento znak informuje funkciu o tom, že začína blok dát od externého zariadenia. Následne sa tieto dáta ukladajú do pomocného zásobníka až do príchodu ukončovacieho znaku "]".

Zároveň sú prijaté dáta porovnávané so znakom "#". Po prijatí tohto znaku je automaticky volaná funkcia *sendConfiguration()*, ktorá odošle kompletnú konfiguračnú tabuľku do počítača.

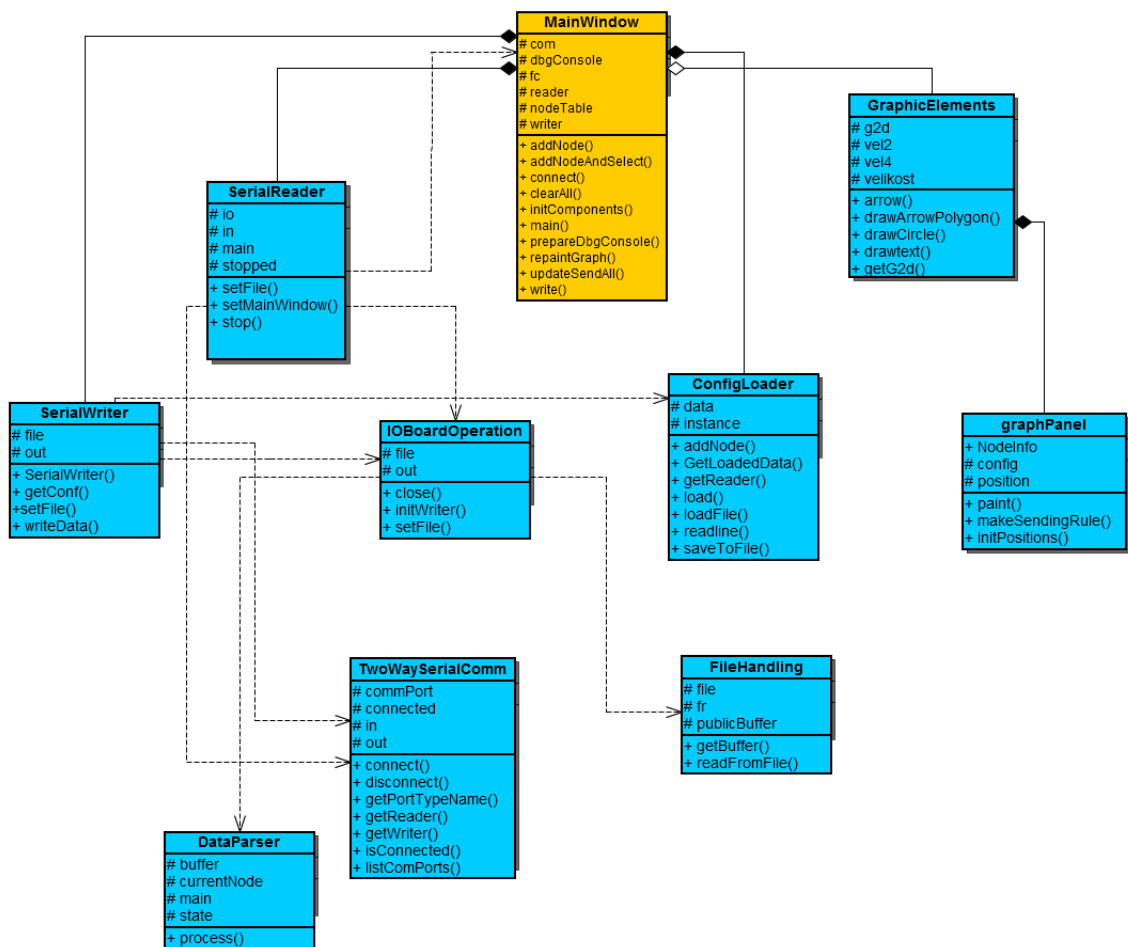
Po uložení dát do dočasného zásobníka, sa zisťuje koľko rámcov prišlo z počítača. Po prijatí len jedného rámca sa zisťuje číslo položky, ktorá sa bude meniť v konfiguračnej tabuľke a následne odosielať konkrétnemu uzlu. Druhá možnosť prijatého počtu rámcov znamená úplnú výmenu dát v konfiguračnej tabuľke. To znamená: prijaté dáta sa rozparsujú, uložia do tabuľky a následne sú jednotlivé položky z tabuľky postupne odosielané jednotlivým uzlom.

### Členenie tried počítačovej aplikácie

Aplikácia je programovaná v grafickom vývojovom prostredí NetBeans IDE 6.9.1 (Build 201011082200) [5]. Na strane počítača sa v aplikácii o prijímanie a odosielanie dát stará trieda *TwoWaySerialComm.java*. Štruktúra aplikácie - *CommunicationInterface.jar* je členená do následovných blokov tried:

1. **triedy pre komunikáciu cez USART:** SerialReader.java, SerialWriter.java, TwoWaySerialComm.java<sup>3</sup>
2. **triedy pre parsovanie dát:** DataParser.java, ConfigLoader.java
3. **triedy pre ukladanie a čítanie dát do súboru:** FileHandling.java
4. **triedy pre tvorbu grafického rozhrania:** graphPanel.java, GraphicElement.java,
5. **hlavná trieda:** MainWindow.java

Rozloženie a interakcia jednotlivých tried je znázornená na obrázku č.4.8.



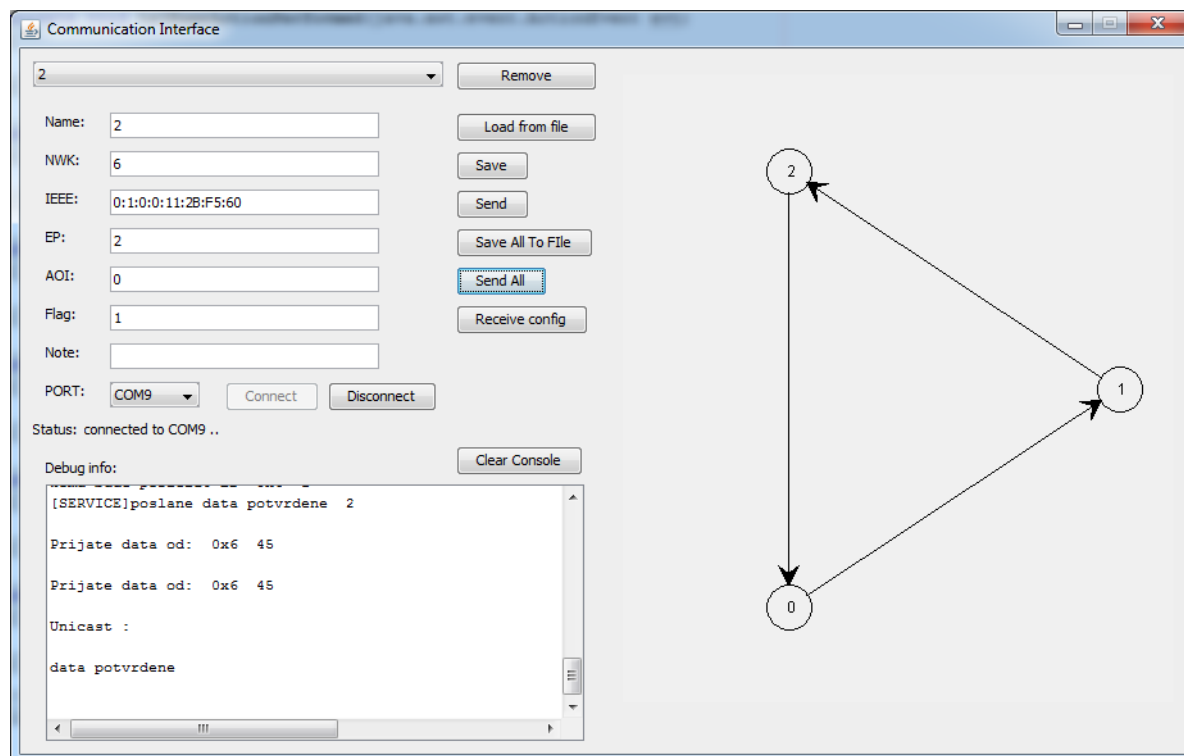
Obr. 4.8: Class diagram Aplikačnej úlohy

Hlavná trieda MainWindow tvorí hlavné vlákno, z ktorého sa spúšťajú ostatné procesy. Pomocou tried 2 komunikuje s triedou na spodnej vrstve, ktorej úlohou je

<sup>3</sup>Použitá knižnica pre real-time obsluhu linky RXTXcomm.jar[18]

prenos dát a koordinátorovi a nastavenie parametrov komunikácie. Ku grafickému zobrazovaniu používa triedy 4 a implementuje grafické prvky. V aplikácii je rovnako ako aj u koordinátora siete naprogramované parsovanie dát. O ukladanie konfigurácie sa starajú triedy 3.

## 4.2.2 Popis Aplikačného Prostredia (GUI)



Obr. 4.9: Grafické rozhranie aplikácie

Grafické prostredie slúži na nastavenie smerovej komunikácie zariadení a pozostáva z troch častí:

1. Nastavovanie parametrov - ľavá horná polovica
2. výpis komunikácie do konzoly - ľavá spodná polovica
3. zobrazovacia časť smerovania toku dát od jednotlivých uzlov - pravá časť okna

Na ukážke č.4.9 môžeme vidieť príklad nastavenia siete.

### Význam zobrazovacích a ovládacích prvkov

V prvej časti možno vidieť sadu tlačidiel a textových polí, ktoré slúžia na nastavenie parametrov jednotlivým uzlom. Ich význam je nasledovný:

- **combobox:** zobrazuje postupne pripájajúce sa zariadenia
- textové pole **Name:** zobrazuje názov pripojeného uzlu(jeho poradové číslo po pripojení). Je totožné s číslom zobrazujúcim sa v combobox.
- textové pole **NWK:** zobrazuje sieťovú adresu pripojeného uzlu
- textové pole **IEEE:** zobrazuje MAC adresu pripojeného uzlu
- textové pole **EP:** zobrazuje endpoint pripojeného uzlu
- textové pole **AOI:** zobrazuje sieťovú adresu, na ktorú bude zariadenie posielat dáta
- textové pole **flag:** rezervované pole pre budúce použitie
- textové pole **Note:** poznámka pre užívateľa. Tu sa môže nastaviť napr. umiestnenie zariadenia v objekte
- tlačidlo **Remove:** slúži na vymazanie kompletného záznamu jednej položky z GUI
- tlačidlo **Load From File:** slúži na načítanie uloženej konfigurácie zo súboru na disku
- tlačidlo **Save:** slúži na uloženie konkrétnej položky uzlu do pamäte
- tlačidlo **Save All To File:** slúži na uloženie kompletnej konfigurácie uzlov do súboru
- tlačidlo **Send All:** Odošle kompletnú konfiguráciu do ZC
- tlačidlo **Receive Config:** slúži na vyžiadanie kompletnej konfigurácie uloženej v ZC
- combobox **Connect:** slúži na pripojenie sa k ZC pomocou sériového portu.
- tlačidlo **Disconnect:** slúži na odpojenie sa od pripojeného uzlu na danom sériovom porte
- tlačidlo **Clear Console:** slúži na vymazanie výpisu v konzole

Druhú časť tvorí konzola(debugovacie okno), do ktorej je presmerovaný výpis chybových hlások a komunikačného toku v sieti. Dáta z procesnej komunikácie sú znázornené identifikátorom "[SERVICE]", pokiaľ dáta z prevádzky sú bez identifikátora.

Tretia časť, je čisto grafické znázornenie smeru toku dát po nastavení uzlu. Uzol tvorí kruh, v ktorom je jeho číslo. Počet kruhov je vytvorený na základe počtu uzlov. Uzle postupným pridávaním vytvárajú kružnicu. Tok dát od jednotlivých zariadení je znázornený smerovými šípkami. Maximálny počet uzlov je v aplikácii obmedzený na 20, z dôvodu prehľadnosti.

## 4.3 Správa Siete z Pohľadu Užívateľa

Táto sekcia predstavuje manuál pre užívateľa, pri zostavovaní siete, jej konfigurácii a možnej poruche. Sled jednotlivých úkonov je znázornený na sekvenčnom diagrame v prílohe E.1.

Po rozvrhnutí rozloženia zariadení v objekte začneme s ich fyzickou prípravou. To zahŕňa rozbalenie zariadení, vloženie batérií.

1. Vo firmware pre ZC nastavíme počet zariadení, ktoré sa v sieti predpokladá.
2. Nasleduje nahratie firmvéru do všetkých zariadení v sieti, pričom rozlišujeme dva druhy firmvéru - coordinator.srec a endDevice.srec.
3. Pomocou PC, na ktorom beží konfiguračná aplikácia, ďalej len aplikácia, sa pripojíme pomocou tlačítka connect ku sériovému portu k ZC.
4. Po nahratí firmvéru, začneme zariadenia pripojovať postupne do siete tak, aby bolo dostatok času na ich rozlíšenie, ktoré kam patrí. V aplikácii možno pozorovať pridávanie uzlov do siete postupným rozširovaním tabuľky. Počet pripojených zariadení vidíme v combobox-e.
5. Po zapnutí a pripojení všetkých zariadení nasleduje konfigurácia zariadení. Každému zariadeniu(uzlu) prislúcha tabuľka hodnôt, ktoré môžeme vidieť v prvej časti aplikácie. Položky Name, NWK, IEEE sú nemenné a sú získané z jednotlivých uzlov. Konfiguračné položky sú: EP, AOI, flag, Note. Je teda potrebné ich vyplniť. Povinné polia sú cieľová adresa a komunikačný endpoint. Položka Note je len poznámka k danému uzlu, z dôvodu prehľadnosti. Pri každom vyplnení položiek je potrebné ich uloženie. K tomu slúži tlačidlo Save. Následne po jeho zmačnutí prebieha kontrola správnosti vyplnenia údajov. To znamená: AOI musí byť zo zoznamu platných adries, teda nemôžeme zadať neexistujúcu adresu, EP je potrebný pre komunikáciu medzi aplikáciami v definovanom profile. Ďalej sa v pravej polovici obrazovky vykresluje počet uzlov spolu s tokom ich komunikácie. Po riadnom vyplnení všetkých položiek od každého uzla môžeme konfiguráciu poslať do ZC, ktorý roz distribuje zadané položky po sieti. Pomocou tlačidla Save All To File si môžeme vyplnenú konfiguráciu uložiť do súboru na disk, a v inom prípade ju zas načítať pomocou tlačidla Load From File.
6. Po odoslaní dát ZC je už sieť nakonfigurovaná a schopná prevádzky. Pri vzniku chyby typu krátkodobého výpadku ZC automaticky pridáva zariadenie do siete a posiela mu konfiguračné údaje.
7. Pomocou aplikácie môžeme v sieti prevádzať následovne úpravy: Pomocou tlačidla Receive Config získame kompletnú konfiguráciu, uloženú v ZC. Upraviť môžeme jeden záznam a odoslať ho pomocou tlačidla Send, ktoré pošle konfiguráciu konkrétneho uzlu do ZC, ktorý si tento záznam upraví a pošle

nové údaje uzlu.

## 4.4 Výsledky časov pripojenia sa zariadení do siete pri výpadku

počet meraní	1	2	3	4	5	6	7	8	čas t[ms]
1 ZED	3687	3656	3656	3656	3233	3687	3718	3656	3619
1. ZED	3687	3728	3687	3656	3687	3233	3718	3656	3632
2. ZED	3656	3233	3687	3686	3728	3656	3656	3687	3624
1+2 ZED priemer	3671,5	3480,5	3687	3671	3707,5	3444,5	3687	3671,5	3628
									priemer
p.uzlov/p.paketov	35	35	30	34	33	35	35	35	34

Tab. 4.1: Tabuľka meraní doby pripojenia sa zariadení pri priamej viditeľnosti, vzdialenosť 20cm

počet meraní	1	2	3	4	5	6	7	8	čas t[ms]
1 ZED	3687	3687	3656	3750	3656	3687	3656	3687	3683
1. ZED	3687	3687	3687	3687	3687	3687	3687	3750	3695
2. ZED	5531	3687	3687	3687	3750	3656	3750	3687	3929
1+2 ZED priemer	4609	3687	3687	3687	3718	3671	3718	3718	3812
									priem. počet
p.uzlov/p.paketov	35	30	35	35	35	30	34	35	34

Tab. 4.2: Tabuľka meraní doby pripojenia sa zariadení pri prekážke(hrúbka steny 15cm, vzdialenosť 5m)

Namerané výsledky pripojenia sa zariadení v prípade výpadku sú zaznamenané v jednotlivých tabuľkách. Možné druhy obmedzenia spolu z výsledkami sú rozdelené do týchto tabuliek:

1. Tabuľka č.4.1 : Zariadenia majú na seba priamu viditeľnosť. Ich vzdialenosť činí 20cm. Časy pripojenia sa pohybovali okolo 3,6s. Množstvo poslaných paketov bolo v priemere 34.
2. Tabuľka č.4.2 : Zariadenia pri meraní oddelovala betónová stena o hrúbke 15cm. Čas pripojenia sa mierne zvýšil.
3. Tabuľka č.4.3 : Medzi zariadeniami bola 1x betónová stena o hrúbke 15cm a dvojce dvere. Priemerná doba pripojenia zariadení bola 4s. Počet prenesených paketov tu dosahoval najnižšiu hodnotu (30) vzhľadom k prekážkam.



počet meraní	1	2	3	4	5	6	7	8	čas t[ms]
1 ZED	3396	3687	3687	5500	3656	3687	3687	3718	3877
1. ZED	3656	6062	3656	3687	3687	3687	3687	3718	3980
2. ZED	3750	3687	5500	5500	3656	5500	7340	3781	4839
1+2 ZED priemer	3703	4875	4578	4594	3672	4594	5514	3750	4410
									priem. počet
p.uzlov/p.paketov	23	30	30	30	35	35	28	29	30

Tab. 4.3: Tabuľka meraní doby pripojenia sa zariadení pri prekážke(hrúbka steny 2x15cm, vzdialenosť 15m)

počet meraní	1	2	3	4	5	6	7	8	čas t[ms]
1 ZED	3656	3656	3656	3656	3656	3656	3656	3656	3656
1. ZED	3656	3687	3656	3718	3233	3656	3656	3656	3615
2. ZED	3656	3656	3687	3656	3656	3656	3718	3656	3668
1+2 ZED priemer	3656	3671,5	3671,5	3687	3444,5	3656	3687	3656	3641
									priem. počet
p.uzlov/p.paketov	30	30	35	35	34	30	30	30	32

Tab. 4.4: Tabuľka meraní doby pripojenia sa zariadení pri prekážke(hrúbka steny 15cm, vzdialenosť 25m)

4. Tabuľka č.4.4 : Posledné meranie bolo pri prenose dát z prízemí na poschodie, čo predstavuje výšku 24m a jednu stenu. Čas pripojenia činí v priemere 3,6s. Počet prenesených paketov za stanovenú dobu bol 32.

## Výsledok meraní

Merania na základe vzdialenosti zariadení a prekážok medzi nimi som rozdelil do tabuliek, ktoré sú zobrazené v kapitole "Výsledky časov pripojenia sa zariadení do siete pri výpadku". testovanie prebiehalo pri vypnutom kryptovaní dát a bez APS potvrdzovania.

Zariadenia dosahovali v najhoršom prípade (vzdialenosť 15m, prekážky - 2x dvere a 1x stena) dobu pripojenia 4,4s. Vzhľadom k pripájaniu sa zariadení pri priamej viditeľnosti je výsledok ich pripájania vyhovujúci. Z tabuliek možno vidieť že pri zväčšujúcej sa vzdialenosti rastie i doba pripájania a znižuje sa počet prenesných paketov za jednotku čas, v mojom prípade 1s.

## 5 ZÁVER

Diplomová práca pojednáva o možnosti návrhu algoritmov pre jednotlivé zariadenia v ZigBee v sieti s cieľom konfigurácie a nastavenie parametrov pre smerovanie toku dát, a tiež možnosť automatického pripojenia zariadenia a uvedenia do prevádzky v prípade krátkodobého výpadku, ako napríklad zlyhania dodávky napájania do zariadenia. Príkladom môže byť vybitie batérií, ich výmena a opätovné spustenie zariadenia. Hlavným kritériom bolo minimalizovať potrebu zásahu ľudského faktoru pri spomínanom výpadku a tiež vytvorenie užívateľského rozhrania na externom zariadení, v tomto prípade počítač, pre konfiguráciu siete pred zahájením prevádzky a počas prevádzky.

V prvej kapitole je popísaný ZigBee štandard z aplikačného hľadiska. To znamená aké typy správ je možno posilať a ako, spôsob adresácie, typy zariadení a ich zrovnanie so štandardom IEEE 802.15.4. Druhá kapitola sa venuje spôsobu párovania (commissioning) zariadení, ich vyhodnoteniu a návrhu riešenia pre svoju aplikáciu. V tretej kapitole je rozobraný návrh siete, na ktorú som postavil algoritmus pre koncové zariadenie ZED, smerovač ZR a koordinátor siete ZC. Je predložená výmena dát v sieti. Dátový tok v sieti možno vidieť na obrázku č. 3.2. Ako prvé ZC sformuje sieť a na túto sieť sa pripájajú ostatné zariadenia. Po každom pripojení nového zariadenia, toto zariadenie odošle koordinátorovi svoje identifikačné údaje, čo sú jeho adresa, MAC adresa a endpoint. Po pripojení PC pomocou sériového rozhrania, ZC automaticky posiela uložené údaje so zariadením na konkrétny port, kde ich načíta vytvorená grafická aplikácia. Po nastavení dôležitých parametrov užívateľom sú konfiguračné údaje poslané späť do ZC, ktorý ich automaticky prepošle konkrétnym zariadeniam. Zariadenia si ukladajú nastavenia a začína režim plnej prevádzky.

Návrh algoritmov pre zariadenia spolu s konfiguračnou aplikáciou je rozobraný v kapitole č.4. Firmvér do ZigBee zariadení sa programoval v jazyku C vo vývojovom prostredí AVR Studio verzii 4.18. Využíval som už vytvorené knižnice poskytnuté firmou ATMEL, ktoré sú dostupné pod názvom BitCloud. Využíval som hlavne funkcie pre prácu so spodnými vrstvami typu ovládanie periférií, štart siete a podobne. Tieto knižnice mi značne urýchlili prácu, vďaka ktorej som sa mohol zamerať len na svoju aplikačnú úlohu. Aplikácia bežiaca na počítači je programovaná v jazyku Java vo vývojovom prostredí NetBeans IDE 6.9.1(Build 201011082200). Naprogramoval som celkovo tri programy, dva tzv. firmware: jeden pre ZC - *coordinator.srec*, jeden spoločný pre ZED a ZR - *endDevice.srec* a jednu grafickú aplikáciu *CommunicationInterface.jar*. Komunikácia medzi ZC a PC prebieha pomocou USART sériového rozhrania. Kapitola rozoberá samotný návrh aplikácie, jednotlivé stavy, v akých sa môžu zariadenia nachádzať. Vytvoril som tiež samostatnú kapitolu pre komunikáciu, kde je rozobraný typ komunikácie, spôsob a komunikačný rámec. V

podkapitole Popis Aplikačného prostredia je popísané grafické rozhranie, význam jednotlivých funkcií tlačidiel a forma znázorňovania. Náзорnejšie pochopenie práce, teda interakciu medzi jednotlivými zariadeniami a užívateľom som zhrnul do podkapitoly 4.3. Je tu detailne popísaná konfigurácia a spôsob zostavenia siete vo forme manuálu. Náзорná ukážka je v prílohe č.E.1. V závere štvrtej kapitoly sú zhrnuté výsledky meraní doby pripájania sa jednotlivých zariadení do existujúcej siete v prípade výpadku, spolu s množstvom prenesených paketov za stanovenú dobu 1s. Meranie som vykonával pri rôznych vzdialenostiach a s rôznym počtom prekážok. Výsledky merania potvrdili, že pri rastúcom počte prekážok a vzdialenosti, klesá počet prenesených paketov - kvôli odozve siete, a tiež narastá doba pripojenia sa. V testovaných podmienkach popísaných v podkapitole 4.4 dosahovali pri priamej viditeľnosti a bez prekážok zariadenia doby pripojenia v priemere 3,6s s počtom prenesených paketov 34. V najhoršom prípade, tj. vzdialenosť 15m pri existencii dvoch stien o hrúbke steny 15cm bola doba pripojenia maximálne 4,4s s počtom prenesených paketov 30. Vzhľadom k zvýšeniu doby o 0,8s pri spomenutých prekážkach je doba opätovného pripojenia sa do siete vyhovujúca.

Diplomovou prácou som získal široký prehľad o možnostiach využitia ZigBee technológie v priemysle i domácej automatizácii. Vzhľadom na komplexnosť danej problematiky som sa zameral na spôsob párovania zariadení (commissioning), pri výpadku zariadenia v sieti a možnosť konfigurácie pomocou počítača. Tento návrh je popísaný v podkapitole Návrh Algoritmu, bod č. 1 a 4. Ako ďalšie možné vyplešenie a rozšírenie, aby vytvorená aplikácia bola čo najviac konkurencie schopná, by bolo vhodné doprogramovať body 2 a 3, ktoré pojednávajú o výmene zariadenia kus za kus, inými slovami povedané nahradenie existujúceho zariadenia za zariadenie nové a tiež o rozšírenie siete o nové zariadenia.

Možnosť riešenia pridania nového zariadenia by som navrhol následovne: Vo firmvéri pre ZC nastaviť maximálny počet zariadení, ktorý dokáže uložiť, vzhľadom k obmedzujúcemu faktoru - veľkosť pamäti implementovanej na zariadení. Na začiatku nastaviť počiatočný stav siete - teda koľko zariadení obsahuje sieť pri spustení. Po nastavení komunikácie a počas prevádzky vzhľadom na požiadavku o navýšenia počtu zariadení v sieti, v položke flag je potrebné informovať pomocou aplikácie na PC ZC o tom že sa bude zvyšovať počet zariadení. Následne sa v ZC prestaví aktuálny počet zariadení v sieti a pripojí sa nové zariadenie. Pri výmene zariadenia za nové stačí asociovať nakonfigurované údaje zo starého zariadenia novému pri výskyte novej MAC adresy.

# LITERATÚRA

- [1] Daintree. [online].  
URL <http://www.daintree.net/index.php>
- [2] Atalum. 2009, [online].  
URL <http://www.atalum.com/>
- [3] Atmel. 2011, [online].  
URL [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)
- [4] Bluetooth. 2011, [online].  
URL [https://www.bluetooth.org/About/bluetooth\\_sig.htm](https://www.bluetooth.org/About/bluetooth_sig.htm)
- [5] NetBeans. 2011, [online].  
URL <http://netbeans.org/community/releases/69/>
- [6] Wi-Fi. 2011, [online].  
URL <http://www.wi-fi.org/organization.php>
- [7] Corporation, A.: *BitCloud User Guide*. 2325 Orchard Parkway San Jose, CA 95131 USA, 09 2011, [online].  
URL [http://www.atmel.com/dyn/products/tools\\_docs.asp?category\\_id=163&family\\_id=676&subfamily\\_id=2124&tool\\_id=4495](http://www.atmel.com/dyn/products/tools_docs.asp?category_id=163&family_id=676&subfamily_id=2124&tool_id=4495)
- [8] Farahani, S.: *ZIGBEE WIRELESS NETWORKS AND TRANSCEIVERS*. ISBN 0750683937, 2008, 360 s.
- [9] Fraile, J. A.; Bajo, J.; Corchado, J. M.; aj.: IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE 1 Applying Wearable Solutions in Dependent Environments.
- [10] Gislason, D.: *ZIGBEE WIRELESS NETWORKING*. newnes, 2008, ISBN 0750685972, 448 s.
- [11] IEEE: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). online, September 2006, [online].  
URL <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>
- [12] IEEE: 2010, [online].  
URL [www.ieee.org](http://www.ieee.org)

- [13] Kim, T.; Kim, D.; Park, N.; aj.: Shortcut Tree Routing in ZigBee Networks.
- [14] Kwon, T. M.; Weidemann, R.: • ZigBee and Mesh Networks • Intersection Simulator • Node Hardware • Experimental Results.
- [15] MeshMetrics: [online].  
URL <http://www.meshnetics.com/dev-tools/zdk900/>
- [16] Microsystems, S.: Java. [online].  
URL [http://java.com/en/download/faq/whatis\\_java.xml](http://java.com/en/download/faq/whatis_java.xml)
- [17] Pekhteryev, G.; Sahinoglu, Z.; Orlik, P.; aj.: Image transmission over IEEE 802.15.4 and zigbee networks. In *In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS, 2005, s. 23–26.*
- [18] RXTX: Marec 2011, [online].  
URL [http://rxtx.qbang.org/wiki/index.php/Main\\_Page](http://rxtx.qbang.org/wiki/index.php/Main_Page)
- [19] Sahinoglu, Z.; Orlik, P.; Zhang, J.; aj.: Reliable Broadcast in ZigBee Networks. 2005.
- [20] Ľubomír, M.: Zigbee - aplikačná vrstva. Marec 2010, [online].  
URL <http://www.aminawsn.org/cs/tutorial/ieee802154-zigbee/100-zigbee-aplikacna-vrstva-apl.html>
- [21] Ľubomír, M.: Zigbee - topológia Hviezda, Strom, Mesh. Marec 2010, [online].  
URL <http://www.aminawsn.org/cs/tutorial/ieee802154-zigbee/96-zigbee-topologie.html>
- [22] Wheeler, A.; Corporation, E.: TOPICS IN AD HOC AND SENSOR NETWORKS Commercial Applications of Wireless Sensor Networks Using ZigBee.
- [23] Yoo, G.; Lee, E.: Self-Healing Methodology in Ubiquitous Sensor Network.

# ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

DSP	číslicové spracovanie signálov – Digital Signal Processing
ZC	ZigBee koordinátor – ZigBee Coordinator
ZED	ZigBee koncové zariadenie – ZigBee End Device
ZR	ZigBee smerovač – Router
Wi-Fi	Obchodná značka Wi-Fi Aliancie
OUI	Organizational Unique Identifier
OEM	Original Equipment Manufacturer
NWK	Network Layer - Sieťová vrstva
PC	počítač – Personal Computer
PDA	mobilné zariadenie – Personal Digital Assistant
O-QPSK	Offset-Qadrature Phase-Shift Keying
DSSS	Direct Sequence Spread Spectrum
CSMA-CA	Carrier Sense Multiple Accces Collision Avoidance
CRC	Cyclic Redundant Check
FCS	Frame Checksum
AES	Advanced Encryption Standard
GHz	gigahertz
kbps	kilobit per second
ZDO	ZigBee Device Object
ZDP	ZigBee Device Profile
kbps	kilobit per second
ZCL	ZigBee Cluster Library
PAN	Personal Area Network - Osobné siete
ID	Identifier - identifikátor

MAC Identifiers

HA Home Automation

NVM Non-volatile memory

P2P peer-to-peer

FFD full-function device

RFD reduced-function device

IDE Integrované vývojové prostredie - Integrated Development Environment

MHZ megahertz

RF/MCU Radio Frequency / Microcontroller

RF Radio Frequency - Rádiová frekvencia

LED Luminiscenčná dióda - light-emitting diode

USB Universal Serial Bus

API Programovacie rozhranie - Application Programming Interface

HAL Hardware Abstraction Layer

BSP Board Support Package

APS Application Support Sublayer

HWD hardvér - Hardware

IRQ Interrupt Request

SPI Serial Peripheral Interface

USART Universal Synchronous/Asynchronous Receiver/Transmitter

$I^2C$  Inter-Integrated Circuit

UID Unique Identification Number

MCU Mikrokonktrolér - Microcontroller

UART Universal Asynchronous Receiver/Transmitter

1-Wire 1-Wire

WSN Wireless Sensor Network - Bezdrôtová Senzorická Sieť

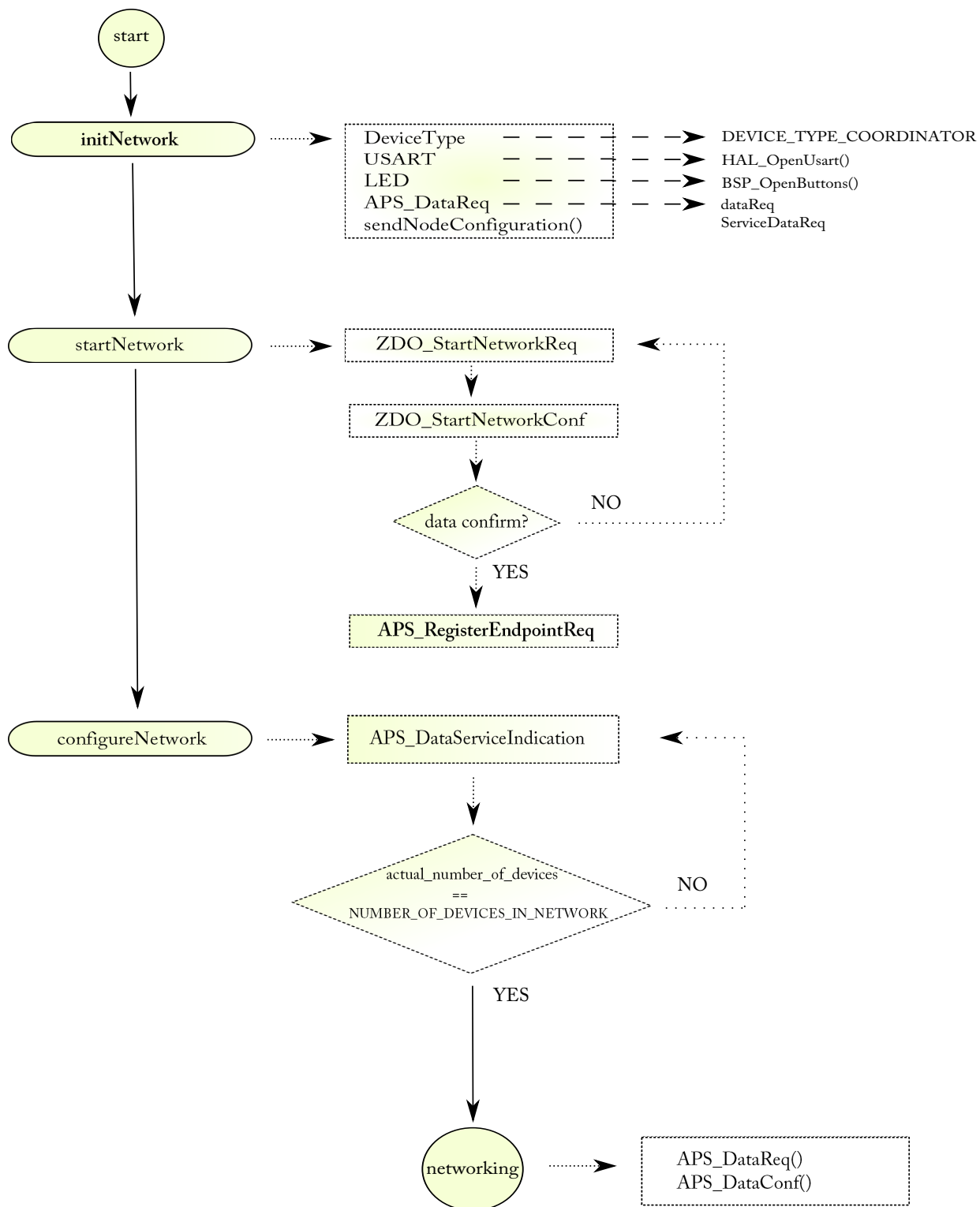
GUI Graphical User Interface - Grafické Rozhranie



# ZOZNAM PRÍLOH

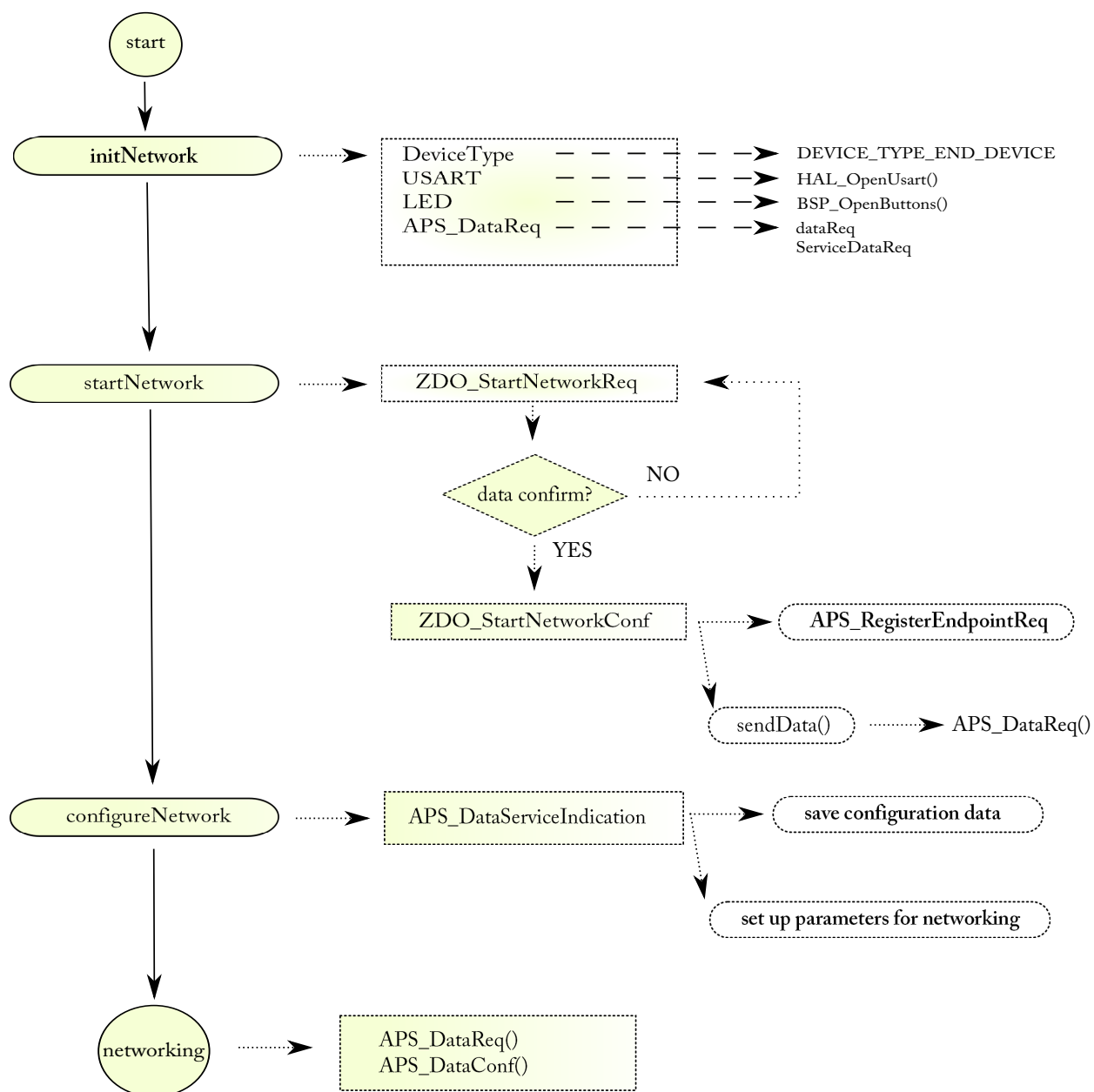
<b>A</b>	<b>Stavový diagram ZC</b>	<b>58</b>
<b>B</b>	<b>Stavový diagram ZED, alebo ZR</b>	<b>59</b>
<b>C</b>	<b>Stavový diagram USART</b>	<b>60</b>
<b>D</b>	<b>Vývojový diagram obsluhy USART</b>	<b>61</b>
<b>E</b>	<b>Sekvenčný diagram pre zostavenie siete</b>	<b>62</b>
<b>F</b>	<b>Obsah Priloženého CD</b>	<b>63</b>
F.0.1	adresár Firmware-C . . . . .	63
F.0.2	adresár Grafická Aplikácia-java . . . . .	63
F.0.3	zip súbor: BitCloud_ZIGBIT_1_10_0.zip . . . . .	63
F.0.4	pdf súbor: Diplomova_praca.pdf . . . . .	63

## A STAVOVÝ DIAGRAM ZC



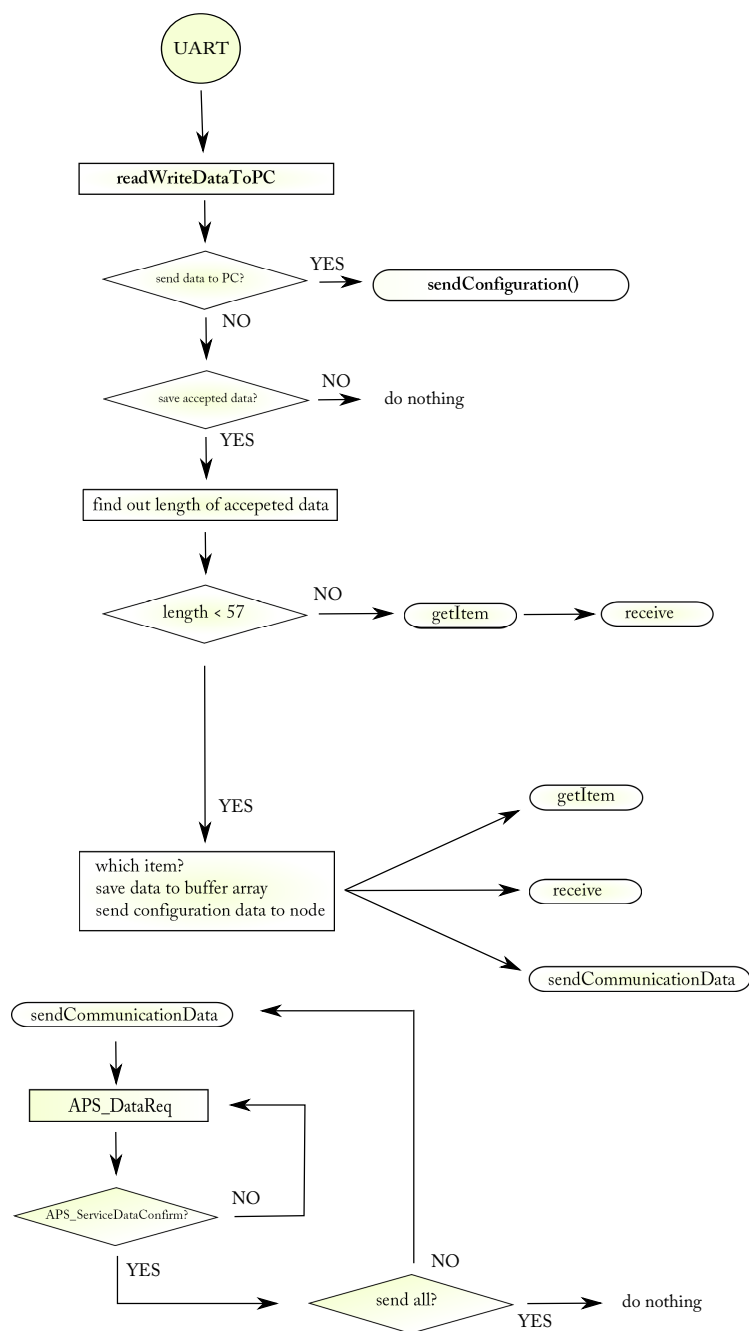
Obr. A.1: Stavový diagram pre Koordinátor

## B STAVOVÝ DIAGRAM ZED, ALEBO ZR



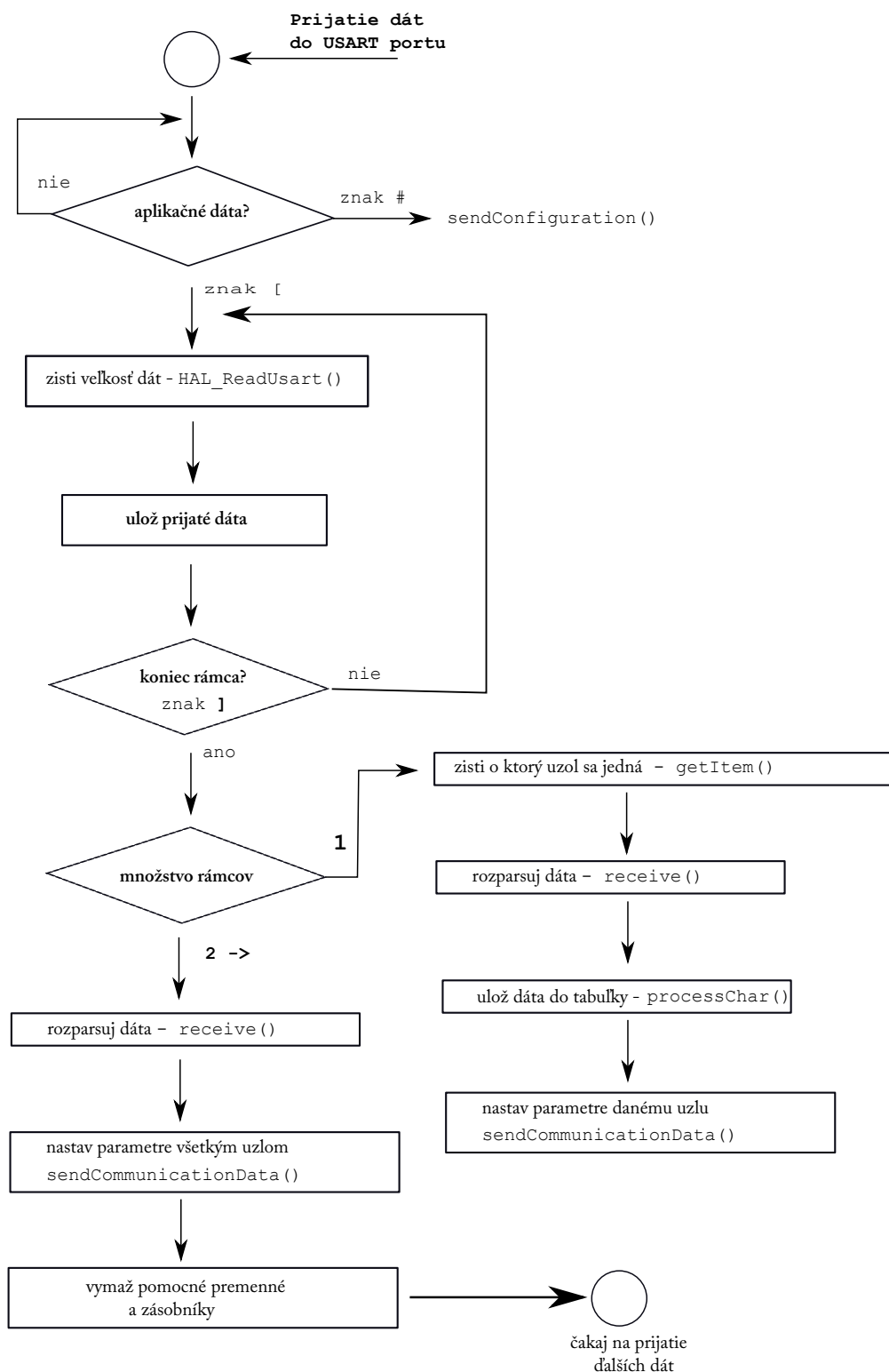
Obr. B.1: Stavový diagram pre Koncové zariadenie a Smerovač

## C STAVOVÝ DIAGRAM USART



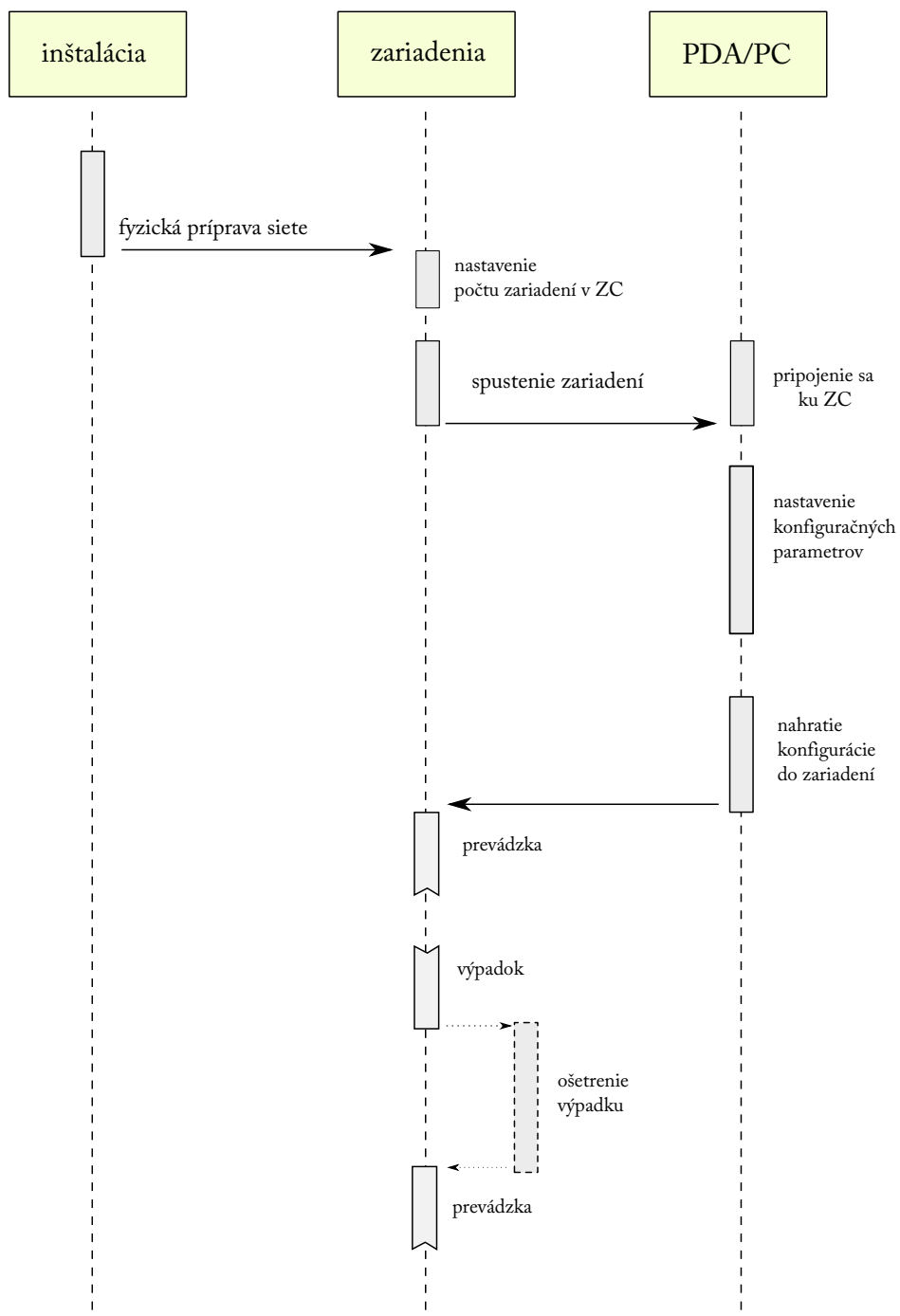
Obr. C.1: Stavový diagram pre Komunikáciu zariadení s externým zariadením určeným na konfiguráciu

## D VÝVOJOVÝ DIAGRAM OBSLUHY USART



Obr. D.1: Vývojový diagram obsluhy USART

## E SEKVENČNÝ DIAGRAM PRE ZOSTAVENIE SIETE



Obr. E.1: Sekvenčný diagram pre zostavenie siete

## **F OBSAH PRILOŽENÉHO CD**

Priložené CD obsahuje

- adresár Firmware-C
- adresár Grafická Aplikácia-java
- zip súbor: BitCloud\_ZIGBIT\_1\_10\_0.zip
- pdf súbor: Diplomova\_praca.pdf

### **F.0.1 adresár Firmware-C**

Obsahuje dva adresáre so zdrojými kódmi v jazyku C a jeden výsledný súbor (\*.srec) pre každé zariadenie.

1. Coordinator
2. EndDevice

### **F.0.2 adresár Grafická Aplikácia-java**

Obsahuje dve adresáre:

1. communicationinterface
2. dist

V 1. adresári sú zdrojové kódy v jazyku Java a knižnicu rxtxSerial.dll pre komunikáciu cez USART. V 2.adresári je výsledná spustiteľná aplikácia, spolu s potrebnými knižnicami.

### **F.0.3 zip súbor: BitCloud\_ZIGBIT\_1\_10\_0.zip**

Knižnice spolu s príkladmi a dokumentáciu BitCloud.

### **F.0.4 pdf súbor: Diplomova\_praca.pdf**

Výsledná Diplomová Práca v elektronickej verzii v pdf formáte.